

Modelos de localización de instalaciones públicas



Pablo Ardoiz García
Trabajo de fin de grado de Matemáticas
Universidad de Zaragoza

Directora: Carmen Galé Pola
Junio, 2022

Resumen

La localización de instalaciones forma parte del conjunto de decisiones estratégicas en muchos problemas del sector logístico empresarial. Estas decisiones se toman a largo plazo e influyen en las decisiones a medio plazo o de planificación y a corto plazo u operativas. Los modelos de ubicación de instalaciones para atender la demanda de usuarios geográficamente dispersos se comenzaron a estudiar en la década de los sesenta en diferentes áreas como la investigación operativa, la geografía económica o la planificación del territorio y siguen muy presentes en la actualidad tanto para la localización de instalaciones públicas como escuelas u hospitales, como para instalaciones del sector privado. En este trabajo fin de grado (TFG) el interés se va a centrar en la localización de instalaciones públicas.

En el capítulo 1 se introducen los modelos de localización de instalaciones y su clasificación según el espacio de decisiones. En estos modelos se toman dos tipos de decisiones que afectan a dónde se ubican las instalaciones y cómo se asignan los usuarios a la instalación. La comparación entre soluciones posibles atiende a diferentes criterios de optimización. En particular, la definición de distancia y criterios relacionados con la satisfacción de los usuarios de gran importancia en el sector público. El capítulo finaliza con una sección en la que se describe la contribución de este TFG.

El capítulo 2 revisa la formulación matemática del modelo más simple de localización y del modelo de la p -mediana, para continuar con variantes más complejas que incluyen capacidad en las instalaciones y el efecto de las restricciones de asignación de los usuarios en ese caso. Además de la asignación única a una instalación o la asignación a múltiples instalaciones, se introducen restricciones de cercanía. Estas restricciones junto con las restricciones de capacidad limitan la factibilidad del modelo y se introducen restricciones que toman en consideración la ruta entre cada usuario y la instalación asignada. Esto supone calcular para cada usuario y ubicación potencial un conjunto de clientes cercanos a la ruta que recorre el usuario para alcanzar la ubicación. Como distancias se han considerado la distancia Euclídea y la distancia Manhattan. Aunque la distancia Manhattan es mayor que la Euclídea, puede ser más realista en el caso de las ciudades. Los modelos de instalaciones se completan con un modelo jerarquizado en el que cada usuario puede tener varios niveles de demanda, así como las ubicaciones pueden tener instalaciones de uno o varios niveles de demanda, según se permita la co-localización o no.

Finalmente, en el capítulo 3 se realiza una experiencia computacional que permite ilustrar cada uno de los modelos introducidos. Para ello se ha generado un ejemplo con 20 ubicaciones potenciales y 20 puntos de demanda. Todos los modelos se han resuelto con el software de optimización de IBM ILOG CPLEX Optimization Studio con licencia académica. En el modelo más simple de localización se analiza a través de varios escenarios el impacto de los costes fijos de apertura de las instalaciones y las alternativas para considerar este tipo de coste. En los modelos de la p -mediana, los escenarios se definen según el valor de p y la distancia considerada. Los modelos de localización con capacidad permiten explorar las diferencias entre las restricciones de asignación única y múltiple, las restricción de asignación a la instalación más cercana o la inclusión de restricciones de asignación a ruta. En el modelo jerarquizado se definen escenarios atendiendo al factor de la co-localización y la restricción sobre el número de instalaciones de cada nivel que pueden abrirse.

Summary

The location of facilities is part of the strategic decision process for many problems in the business logistics sector. These decisions are taken over a long term and influence medium-term/planning and short-term/operational decisions. Models for the location of facilities to attend to the demand of geographically dispersed users, began to be studied in the 1960s in different areas such as operational research, economic geography or territorial planning and are still very present today both for the location of public facilities such as schools or hospitals, as well as for private sector facilities. In this Final Degree Project, the focus will be on the location of public facilities.

Chapter 1 introduces the models of facilities location and their classification according to the decision-making space. In these models, two types of decisions are made that affect where facilities are located and how users are allocated to the facility. The comparison between feasible solutions is based on different optimization criteria. Particularly, the definition of distance and criteria related to user satisfaction is of great importance in the public sector. The chapter ends with a section describing the contribution of this project.

Chapter 2 reviews the mathematical formulation of the simple plant location problem and the p -median model, followed by more complex variants that include facility capacity constraints and the effect of user allocation constraints in this case. In addition to allocation to a single installation or allocation to multiple installations, proximity constraints are introduced. These constraints together with the capacity constraints limit the feasibility of the model and other constraints are introduced that take into consideration the route between each user and each facility. This involves calculating for each user and potential location a set of users close to the route that the user travels to reach the location. Euclidean distance and Manhattan distance have been considered as distances. Although the Manhattan distance is larger than the Euclidean distance, it may be more realistic for cities. The facility models are completed with a hierarchical model in which each user can have several levels of demand, just as locations can have facilities of one or several levels of demand, depending on whether co-location is allowed or not.

Finally, in chapter 3, a computational experiment is carried out to illustrate each of the models introduced. For this purpose, an example has been generated with 20 potential locations and 20 demand points. All models have been solved with IBM ILOG's academically licensed optimization software CPLEX Optimization Studio. In the simple location model, the impact of the fixed costs of opening the facilities and the alternatives to consider this type of cost are analyzed through various scenarios. In the p -median models, the scenarios are defined according to the value of p and the considered distance. The location models with capacity allow exploring the differences between single and multiple allocation constraints, the nearest facility allocation constraints, or the inclusion of route allocation constraints. In the hierarchical model, scenarios are defined based on the co-location factor and the constraint on the number of facilities at each level that can be opened.

Índice general

Resumen	III
Summary	v
1. Modelos de localización de instalaciones	1
1.1. Introducción	1
1.2. El espacio de decisiones en los problemas de localización	1
1.3. Criterios de optimización en la ubicación y asignación	2
1.4. Contribución del TFG	3
2. Formulación matemática de los modelos	5
2.1. Introducción y notación	5
2.2. Modelo simple de localización	5
2.3. Modelo de la p -mediana	6
2.4. Modelo con capacidad en las instalaciones	7
2.5. Restricciones de asignación de usuarios a instalaciones	8
2.6. Modelo jerarquizado	9
2.6.1. Formulación matemática	9
2.6.2. Variantes del modelo jerarquizado	11
3. Aplicación y análisis de los modelos	13
3.1. Introducción	13
3.2. Descripción del problema	13
3.3. Modelos SPLP	14
3.4. Modelos de la p -mediana	15
3.5. Modelos con capacidad	18
3.6. Modelo jerarquizado	22

Capítulo 1

Modelos de localización de instalaciones

1.1. Introducción

En un problema de localización de instalaciones se considera un conjunto potencial de ubicaciones para establecer una instalación y un conjunto de usuarios geográficamente dispersos que han de ser atendidos desde alguna de las instalaciones. El objetivo, en general, consiste en elegir un subconjunto de ubicaciones para situar las instalaciones de forma que todos los usuarios sean atendidos con el menor coste posible. En el caso de la localización de instalaciones públicas, la satisfacción de los usuarios es un aspecto relevante y ha de ser considerado.

Como explican Teixeira y Antunes (2006) [1], durante las últimas décadas, los procesos de localización de instalaciones públicas como escuelas u hospitales se han vuelto cada vez más complejos debido a la consideración de procesos de participación de la población en los que surgen los diferentes puntos de vista e intereses. Por ello, los modelos de localización son herramientas esenciales en la toma de decisiones, especialmente cuando el conjunto de posibles soluciones es grande. Estos modelos han sido estudiados desde la década de los sesenta en diferentes áreas como la investigación operativa, la ingeniería industrial, la geografía económica y la planificación de territorio.

1.2. El espacio de decisiones en los problemas de localización

ReVelle y Eiselt (2003) [3] proponen una primera clasificación en los problemas de localización, que viene dada por el espacio en el que se representa el sistema real bajo estudio. Este espacio puede ser un grafo, o de forma general el espacio d-dimensional \mathbb{R}^d , aunque en la práctica el más utilizado es \mathbb{R}^2 . En el plano los usuarios y las instalaciones se sitúan en cualquier punto y los usuarios pueden moverse libremente hacia las instalaciones. En un grafo los usuarios y las instalaciones se sitúan en los nodos y/o puntos de las aristas del grafo y los usuarios se mueven a lo largo de las aristas, por lo que las distancias se calculan de manera diferente.

En la figura I.1 se muestra un problema en el plano I.1(a) y un problema en un grafo I.1(b). Los puntos azules representan a los usuarios mientras que los puntos rojos representan las instalaciones abiertas y los puntos negros las instalaciones que no se han abierto. En un problema de localización continuo las instalaciones pueden ubicarse en cualquier parte del plano o de la red, mientras que en un problema de localización discreto, las instalaciones solo pueden localizarse en un número finito de ubicaciones potenciales. En este caso, es preciso hacer un estudio previo de cuales son los sitios candidatos para localizar la instalación.

En general, en los problemas de localización continua se considera como espacio de decisión el plano y la distancia euclídea, lo que conduce a problemas de optimización cuadrática. Por el contrario, en los problemas de localización discreta es habitual considerar un grafo como espacio de decisión y el uso de variables binarias en su formulación. Por ello, estos problemas se formulan como problemas de programación entera de naturaleza combinatoria. En este trabajo se va a considerar un problema de localización discreta en el plano (figura I.1(a)).

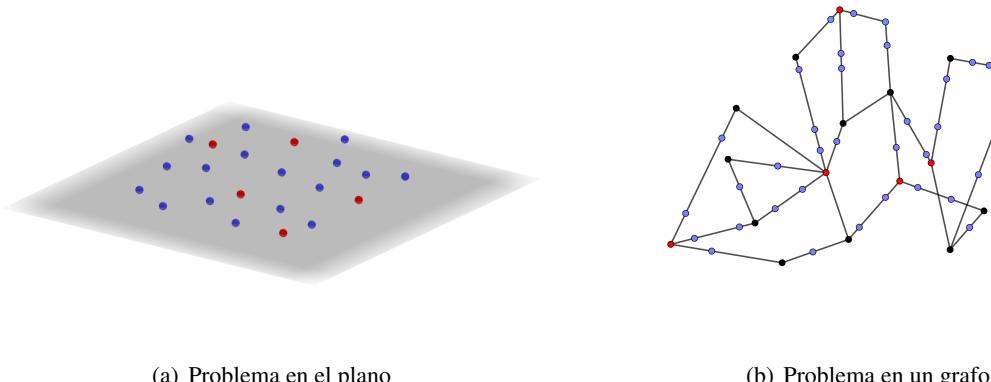


Figura 1.1: Espacios de decisión en problemas de localización

1.3. Criterios de optimización en la ubicación y asignación

En los problemas de localización hay dos tipos de decisión: dónde se ubican las instalaciones y cómo se asignan los usuarios a una instalación y un elemento esencial es la función objetivo o criterio con el que evaluar las soluciones factibles ([3]). Tradicionalmente, se ha buscado que las instalaciones se ubiquen lo más cerca posible de los usuarios pero en este punto, se debe destacar la diferencia entre los objetivos del sector público y del privado.

ReVelle y Eiselt ([3]) explican cómo en el sector privado, los criterios habitualmente son económicos y se busca ubicar las instalaciones en sitios que minimicen los gastos para obtener así mayores beneficios. Por lo general, las empresas bien disponen de un presupuesto para abrir varias instalaciones o bien su interés se centra en ubicar un número de instalaciones dado. Los costes se refieren a los costes fijos de abrir las instalaciones y los costes de producción y transporte a los usuarios. En el caso de empresas que venden sus productos a precio de fábrica, en general, los costes de transporte corren a cargo de los usuarios. Si la empresa paga los costes de transporte, la función objetivo ha de considerar también la minimización de los costes de transporte.

Por el contrario, en el sector público hay que atender a criterios relacionados con la satisfacción de los usuarios. En el caso de instalaciones públicas, como centros de enseñanza o sanitarios al que acuden usuarios de varias poblaciones, la función objetivo ha de atender también a las distancias de los usuarios a dichos centros. Así pues, en el caso de instalaciones públicas cobran más importancia los criterios de optimización en la asignación, que buscan minimizar el coste de asignar un usuario a una instalación determinada. Normalmente esto implica minimizar las distancias de los usuarios a las instalaciones ya que interesa situar dichas instalaciones lo más cerca posible de los usuarios. En unos casos puede interesar minimizar la distancia máxima de un usuario a la instalación y en otros minimizar la distancia media. Otro tipo de objetivos se refieren a la equidad, es decir, se busca que la distancia de los usuarios a las instalaciones sea tan similar como sea posible. Además, se pueden considerar varios criterios simultáneamente utilizando una aproximación multiobjetivo del problema.

ReVelle y Eiselt ([3]) introducen también los problemas de localización de instalaciones “indeseables”. En estos problemas los usuarios quieren estar lo más lejos posible de las instalaciones abiertas. Por ejemplo, en el caso de un vertedero, los usuarios desearán situarlo lo más lejos posible, mientras que a la empresa de recogida de basuras le interesa que la distancia de los usuarios al vertedero no sea mucha para ahorrar en costes de transporte. En este caso, ambos objetivos son conflictivos.

Por tanto, en cualquier problema de localización juega un papel esencial el cálculo de la distancia. Según ReVelle y Eiselt ([3]), la mayoría de distancias utilizadas en \mathbb{R}^2 se derivan de las distancias de

Minkowski . La definición de distancia l_q entre dos puntos (a_i, b_i) y (a_j, b_j) en el plano con $i \neq j$ es:

$$d_{ij} = (|a_i - a_j|^q + |b_i - b_j|^q)^{1/q}$$

Los valores más habituales para q son:

Manhattan Con $q = 1$, la distancia l_1 :

$$d_{ij} = |a_i - a_j| + |b_i - b_j|$$

Euclídea Con $q = 2$:

$$d_{ij} = \sqrt{|a_i - a_j|^2 + |b_i - b_j|^2}$$

Chebyshev Con $q = \infty$:

$$d_{ij} = \max\{|a_i - a_j|, |b_i - b_j|\}$$

La distancia utilizada depende del modelo en el que se aplique. En la práctica se utiliza la distancia Manhattan en problemas referidos a ciudades. En otro tipo de problemas se utiliza la distancia Euclídea o la distancia entre los puntos por carreteras.

1.4. Contribución del TFG

En este TFG se aborda el problema de localización de instalaciones públicas. Se ha revisado la literatura y la formulación matemática del modelo de localización simple y el modelo de la p -mediana, lo que ha facilitado la revisión de modelos más complejos como los modelos de localización con restricciones de asignación a la instalación más cercana o con restricción de asignación de ruta, tal y como se ha denominado. Este tipo de restricciones fueron introducidas por Teixeira y Antunes [1] sin especificar el cálculo de los conjuntos que intervienen en dichas restricciones. En este TFG se han considerado dos definiciones de distancia. La distancia Euclídea proporciona la distancia mínima entre dos puntos, ya que es la longitud del segmento que une los dos puntos. Sin embargo, en el cálculo de rutas mínimas en ciudad no siempre es posible recorrer dicho segmento. En una ciudad, si las calles se distribuyeran según una cuadrícula, la distancia Manhattan o l_1 , proporciona la distancia entre dos puntos, que es la longitud de cualquier camino que une ambos puntos con segmentos horizontales y verticales. Por eso, en ocasiones, en el cálculo de rutas también se trabaja con esta distancia. En la definición de los conjuntos, un conjunto por cada pareja de punto de demanda y ubicación, se han considerado ambas distancias.

Con el lenguaje Python se ha generado un ejemplo ilustrativo con 20 ubicaciones y usuarios, suficientemente grande para ilustrar todos los modelos. En la resolución de los modelos se utiliza el software IBM ILOG CPLEX Optimization Studio. Se ha diseñado una experiencia computacional para evaluar el efecto de parámetros, como los costes fijos o el número de instalaciones abrir; el efecto de la distancia utilizada, la distancia Euclídea o la distancia de Manhattan; el efecto de las restricciones de asignación a la más cercana, la asignación única o múltiple y la asignación de ruta; el efecto de la co-localización y la limitación en la apertura de instalaciones en el modelo jerarquizado.

Capítulo 2

Formulación matemática de los modelos

2.1. Introducción y notación

En este capítulo se introduce la formulación matemática de varios modelos de localización. En primer lugar, en este apartado se presenta la notación común utilizada en los modelos referida a los conjuntos, los parámetros y las variables de decisión. En los siguientes apartados se presentan los tres modelos analizados, desde el más simple al modelo más amplio.

Sea $I = \{1, \dots, n\}$ el conjunto de puntos de demanda. En cada punto $i \in I$ se conoce la demanda u_i o número de usuarios. Sea $J = \{1, \dots, m\}$ el conjunto de ubicaciones potenciales de las instalaciones. Sea f_j el coste fijo de abrir una instalación en $j \in J$. El coste de asignación de un punto de demanda $i \in I$ a una ubicación $j \in J$ se denota por c_{ij} . Este coste puede considerar los costes de viaje desde el punto de demanda a la instalación. Por ejemplo, se puede definir el coste $c_{ij} = u_i d_{ij}$, $i \in I$, $j \in J$, siendo d_{ij} la distancia del punto de demanda $i \in I$ a la instalación $j \in J$ y que tiene en cuenta el número de usuarios de dicho punto.

Las variables de decisión del problema son de dos tipos. Las variables binarias de localización determinan qué instalaciones se abren:

$$y_j = \begin{cases} 1 & \text{se abre la instalación } j \\ 0 & \text{en otro caso} \end{cases} \quad j \in J$$

y variables no negativas de asignación:

$$x_{ij} \geq 0, \quad i \in I, j \in J$$

representan la proporción de demanda del punto $i \in I$ satisfecha por la instalación ubicada $j \in J$.

2.2. Modelo simple de localización

El modelo más básico y del que se derivan modelos más complejos es el problema de localización simple (SPLP, del inglés *Simple Plant Location Problem*). En este problema se determina la ubicación en la que se abren las instalaciones con el fin de minimizar los costes de transporte o de asignación de los usuarios. Las restricciones garantizan que se satisfaga la demanda de todos los usuarios y que los usuarios sean atendidos sólo desde instalaciones abiertas.

La formulación matemática del SPLP es la siguiente:

$$\min_{x,y} \quad \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} f_j y_j \quad (2.1a)$$

$$\text{sujeto a:} \quad \sum_{j \in J} x_{ij} = 1, \quad i \in I \quad (2.1b)$$

$$x_{ij} \leq y_j, \quad i \in I, j \in J \quad (2.1c)$$

$$x_{ij} \geq 0, \quad i \in I, j \in J \quad (2.1d)$$

$$y_j \in \{0, 1\}, \quad j \in J \quad (2.1e)$$

La función objetivo (2.1a) trata de minimizar la suma de los costes de asignación y los costes por la apertura de las nuevas instalaciones. Las restricciones (2.1b) garantizan que para todos los puntos de demanda se satisfaga toda su demanda. Con las restricciones (2.1c) se obtiene que la demanda se asigne sólo a sitios con instalaciones abiertas. Finalmente, las restricciones (2.1d) determinan el signo de las variables de asignación y las restricciones (2.1e) indican que las variables de localización son binarias. En un abuso de la notación, cuando se hace referencia a las variables x se refiere al conjunto $\{x_{ij} : i \in I, j \in J\}$.

En este modelo se considera que los puntos de demanda se pueden atender desde diferentes centros y el número de instalaciones a ubicar se considera en la función objetivo en el término de costes de instalación, ya que se buscará la solución que permita atender toda la demanda y el menor coste de asignación e instalación. En algunos problemas las variables x son también binarias. Esto implica que la demanda del punto un punto $i \in I$, se satisface completamente por una sola instalación.

De forma alternativa, se puede considerar una función objetivo que incluya solo los costes de asignación y los costes de instalación se garantice sean inferiores a un presupuesto dado B añadiendo la restricción:

$$\sum_{j \in J} f_j y_j \leq B$$

En cualquiera de las dos alternativas, el número de instalaciones que se abren se determina como solución al problema. En el siguiente modelo, el número de instalaciones que hay que abrir es un parámetro conocido del modelo.

2.3. Modelo de la p -mediana

En el modelo SPLP, el número de instalaciones atiende a los costes de instalación y no está determinado a priori. En el caso del modelo de la p -mediana, el número de instalaciones que se abren es exactamente p . El problema consiste en calcular el conjunto de p ubicaciones donde localizar esas instalaciones y determinar qué instalación sirve a cada punto de demanda, de manera que los costes se minimicen. Dado que el número de instalaciones está fijado, en la función objetivo del problema no se consideran los costes fijos de apertura.

La formulación matemática del modelo, que utiliza la misma notación que el modelo 2.1, es la

siguiente:

$$\min_{x,y} \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (2.2a)$$

$$\text{sujeto a: } \sum_{j \in J} x_{ij} = 1, \quad i \in I \quad (2.2b)$$

$$x_{ij} \leq y_j, \quad i \in I, j \in J \quad (2.2c)$$

$$\sum_{j \in J} y_j = p \quad (2.2d)$$

$$x_{ij} \geq 0, \quad i \in I, j \in J \quad (2.2e)$$

$$y_j \in \{0, 1\}, \quad j \in J \quad (2.2f)$$

En este modelo, tanto la función objetivo (2.2a) y las restricciones (2.2b), (2.2c), (2.2e) y (2.2f) coinciden con el modelo (2.1). La restricción (2.2d) garantiza que se abren exactamente p instalaciones.

En los modelos SPLP y de la p -mediana, en la solución óptima, los puntos de demanda están completamente atendidos por la instalación abierta más cercana o de menor coste de asignación. Esto sucede porque, como no hay limitaciones de capacidad, dividir la demanda de un punto en varias instalaciones incrementa el coste de asignación frente a ser atendido por la instalación de menor coste de asignación. A continuación, se van a presentar modelos en los que las instalaciones tienen capacidad.

2.4. Modelo con capacidad en las instalaciones

Este modelo es una extensión del SPLP (2.1) en el que se añaden restricciones de capacidad a las instalaciones. Cada instalación puede ser abierta o no. En el caso de ser abierta, la capacidad mínima se refiere al mínimo número de usuarios que han de ser asignados y la capacidad máxima, al número máximo de usuarios que pueden ser atendidos. Sean b_j y B_j el mínimo y el máximo de capacidad de una instalación en el sitio $j \in J$. La formulación del modelo es la siguiente:

$$\min_{x,y} \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} f_j y_j \quad (2.3a)$$

$$\text{sujeto a: } \sum_{j \in J} x_{ij} = 1, \quad i \in I \quad (2.3b)$$

$$x_{ij} \leq y_j, \quad i \in I, j \in J \quad (2.3c)$$

$$\sum_{i \in I} u_i x_{ij} \geq b_j y_j, \quad j \in J \quad (2.3d)$$

$$\sum_{i \in I} u_i x_{ij} \leq B_j y_j, \quad j \in J \quad (2.3e)$$

$$x_{ij} \geq 0, \quad i \in I, j \in J \quad (2.3f)$$

$$y_j \in \{0, 1\}, \quad j \in J \quad (2.3g)$$

Como en modelos anteriores las restricciones (2.3c) y (2.3b) garantizan que la demanda se satisface y es solo atendida por instalaciones abiertas. Las nuevas restricciones (2.3d) y (2.3e) garantizan que el mínimo y el máximo de capacidad de cada instalación se verifique. En este modelo (2.3), al no incluir la restricción (2.2d), el número de instalaciones que se abren queda determinado por la solución.

2.5. Restricciones de asignación de usuarios a instalaciones

En esta sección se analizan diferentes restricciones de asignación de los usuarios a las instalaciones. La demanda de un punto puede ser atendida por varias instalaciones o por una única instalación. En los modelos anteriores las variables de asignación no negativas (2.1d), (2.2e), (2.3f) permiten atender cada punto de demanda por varias instalaciones. En el caso que interese que la asignación sea a una única instalación, se sustituyen dichas restricciones por las siguientes:

$$x_{ij} \in \{0, 1\}, \quad i \in I, j \in J \quad (2.4)$$

Texeira y Antunes ([1]), junto con la restricción de que cada punto de demanda sea atendido por una única instalación, introducen otro tipo de restricción de asignación en el modelo de capacidad. Estas son las restricciones de cercanía y garantizan que la demanda de cada punto sea atendida por la instalación abierta más cercana:

$$\sum_{k \in J | d_{ik} \leq d_{ij}} x_{ik} \geq y_j, \quad i \in I, j \in J \quad (2.5)$$

Las restricciones (2.5) se verifican siempre cuando para una instalación $j \in J$ la decisión sea no abrir, $y_j = 0$. En el caso de una instalación abierta $j \in J$ con $y_j = 1$, entonces, para cualquier cliente $i \in I$:

$$\sum_{k \in J | d_{ik} \leq d_{ij}} x_{ik} \geq 1, \quad i \in I$$

lo que garantiza que será asignado a j o a una instalación k situada a distancia menor o igual que la distancia de i a j . Es decir, los usuarios de un punto de demanda serán atendidos por la instalación abierta más cercana.

Las restricciones de capacidad, la asignación de cada punto de demanda a una única instalación garantizando que es la más cercana puede provocar que el problema no sea factible. Dada una instalación $j \in J$, si la suma de la demanda de los puntos de demanda que tienen dicha instalación como la más cercana, es superior a la capacidad máxima B_j , esa instalación no se puede abrir porque no hay ninguna solución factible al problema (2.3) junto con las restricciones (2.4) y (2.5) con $y_j = 1$. De este modo, es posible que no haya ninguna solución factible, porque la asignación a la planta más cercana provoque que haya plantas en las que superaría la capacidad máxima y en otras no se alcance la capacidad mínima.

En la búsqueda de la factibilidad se puede relajar alguna de las restricciones. Los modelos con capacidad (2.3) no incorporan, en general, de forma simultánea las restricciones de asignación única y asignación más cercana. Sin embargo, en un contexto de instalaciones públicas, hay que evitar que los usuarios de un mismo punto de demanda puedan asignarse a distintas instalaciones o asignarse todos a la misma instalación lejana pero habiendo instalaciones más cercanas o que el camino a la instalación asignada pase por un punto de demanda asignado a otra instalación. Ante estas situaciones, las soluciones son difícil de explicar a los usuarios, por lo que difícilmente será implementada en la práctica.

Texeira y Antunes (2006) ([1]) sugieren introducir restricciones de asignación de ruta en lugar de las restricciones de cercanía (2.5). Estas restricciones garantizan que si se asigna un punto de demanda a una instalación j , todos los centros “cerca” de la ruta recorrida por los usuarios para llegar a j , también deben asignarse a j :

$$\sum_{k \in P_{ij}} x_{kj} \geq |P_{ij}| x_{ij}, \quad i \in I, j \in J \quad (2.6)$$

donde P_{ij} es el subconjunto de puntos de demanda que están cerca del camino más corto entre el punto de demanda i y la instalación j y por $|P_{ij}|$ se denota el cardinal de dicho conjunto.

Las restricciones (2.6) garantizan que si $x_{ij} = 1$, todos los puntos de demanda en el conjunto P_{ij} serán asignados a la instalación j . De forma alternativa, se puede utilizar el siguiente conjunto de restricciones:

$$x_{kj} \geq x_{ij}, \quad i \in I, j \in J, k \in P_{ij} \quad (2.7)$$

La construcción de los conjuntos P_{ij} depende de la aproximación que se utilice para determinar la cercanía de los usuarios a las rutas de otros usuarios y del modelo en el que se aplique. En este TFG se han considerado dos métodos para la construcción de dichos conjuntos.

El primer método se utiliza en los modelos de localización en los que se considera la distancia Euclídea. Dado un punto de demanda $i \in I$ y una ubicación $j \in J$, el conjunto P_{ij} contiene aquellos puntos de demanda $k \in I$ cuya distancia Euclídea a la ruta que recorre el usuario i para llegar a la ubicación j es menor o igual que un cierto ϵ . En la figura 2.1(a), todos los puntos de demanda situados en el área verde pertenecen al conjunto P_{ij} .

El segundo método se utiliza en los modelos de localización en los que el coste se calcula en relación con la distancia Manhattan. En estos modelos no hay una ruta única entre el cliente i y la ubicación j con la mínima distancia. Dado un punto i en el plano, el conjunto de puntos situados a una distancia Manhattan menor o igual a ϵ se sitúan en un rombo cuyas diagonales miden 2ϵ e intersecan en i . Tomando como referencia todas las rutas, en la figura 2.1, los puntos de demanda situados en el área verde pertenecen al conjunto P_{ij} .

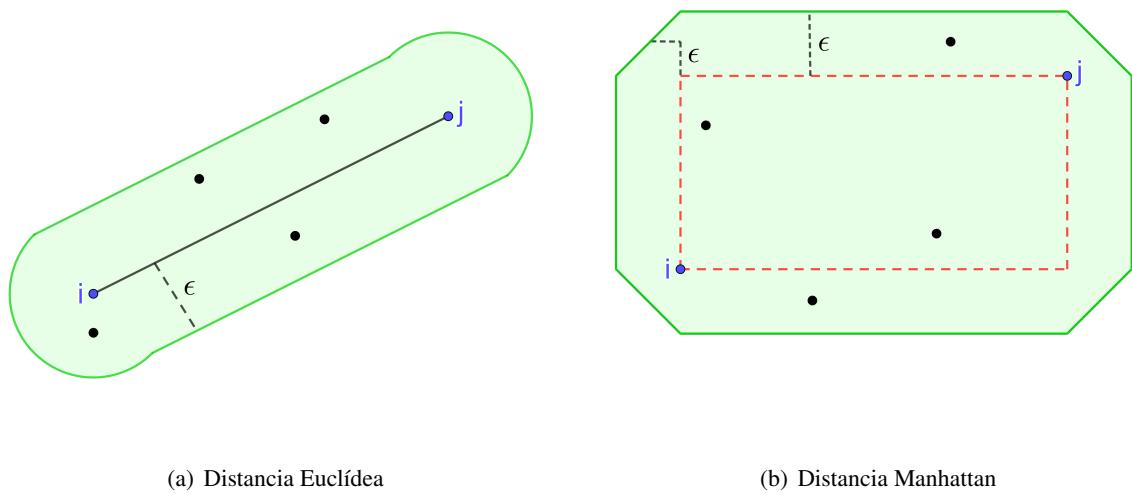


Figura 2.1: Definición de los conjuntos P_{ij} , $i \in I$, $j \in J$

2.6. Modelo jerarquizado

En esta sección se estudia una variante del modelo con restricciones de capacidad (2.3). En lugar de considerar un tipo de demanda único en cada punto de demanda se introducen varios niveles de demanda y por tanto, varios tipos de instalaciones. Un ejemplo clásico de la aplicación de este modelo es la localización de instalaciones públicas sanitarias que pueden ser consultorios, centros de salud y hospitales que ofrecen diferentes servicios. El tipo de servicio que se proporciona en cada instalación es diferente, siendo el hospital el nivel superior y el consultorio el nivel inferior.

2.6.1. Formulación matemática

Sea el conjunto $S = \{1, \dots, l\}$ de niveles de demanda y niveles de instalaciones. Se introduce una jerarquía en las instalaciones, de forma que, una instalación de nivel $s \in S$ puede atender la demanda cuyo nivel se sitúe en el conjunto $\{1, \dots, s\}$. En cada punto de demanda $i \in I$, el valor u_{is} representa la

demandas de nivel $s \in S$ en dicho punto. Dada una instalación $j \in J$ de nivel $s \in S$, la capacidad mínima y máxima de atención a usuarios de nivel $r \in S$ con $r \leq s$ es b_{jr} y B_{jr} , respectivamente. Para cada instalación de nivel $s \in S$ abierta en $j \in J$, f_{js} representa los costes fijos de apertura.

En el modelo se definen adicionalmente el siguiente conjunto y parámetros:

- $J_s^0 \subset J$ conjunto de ubicaciones en las que ya existen instalaciones de nivel $s \in S$
- p_s : número máximo de instalaciones de tipo s que se pueden abrir, $s \in S$
- q_s : número máximo de instalaciones de tipo s que se pueden cerrar, $s \in S$
- D_s : distancia máxima que recorren los usuarios para acceder a una instalación de nivel s , $s \in S$.

En este modelo, las variables de decisión determinan la instalación que proporciona la atención a cada nivel de demanda del punto de demanda. La asignación ha de realizarse a instalaciones que puedan garantizar ese nivel de atención. En la formulación propuesta se relaja la hipótesis de asignación única, por ello las variables de asignación son continuas. Las variables de decisión son:

$$x_{ijs} \geq 0 \quad \text{fracción de demanda de nivel } s \text{ del punto de demanda } i \text{ que se satisface por la instalación } j, i \in I, j \in J, s \in S$$

$$y_{js} \in \{0, 1\} \quad \text{toma el valor 1 si se localiza una instalación de nivel } s \text{ en } j, j \in J, s \in S$$

$$z_{jst} \geq 0 \quad \text{demanda de nivel } s \text{ satisfecha por la instalación ubicada en } j \text{ de nivel } t, j \in J, s, t \in S, s \leq t$$

Una hipótesis adicional en este modelo consiste en permitir que en una ubicación se puedan establecer instalaciones de varios niveles, lo que se denomina como co-localización. Esta posibilidad favorece que se verifiquen las restricciones de capacidad máxima. La formulación matemática del modelo jerarquizado es:

$$\min \sum_{i \in I} \sum_{j \in J} \sum_{s \in S} d_{ij} u_{is} x_{ijs} + \sum_{j \in J} \sum_{s \in S} f_{js} y_{js} \quad (2.8a)$$

$$\text{sujeto a: } \sum_{j \in J} x_{ijs} = 1, \quad i \in I, s \in S \quad (2.8b)$$

$$x_{ijs} \leq \sum_{t \in S \mid t \geq s} y_{jt}, \quad i \in I, j \in J, s \in S \quad (2.8c)$$

$$\sum_{t \in S \mid t \geq s} z_{jst} = \sum_{i \in I} u_{is} x_{ijs}, \quad j \in J, s \in S \quad (2.8d)$$

$$\sum_{s \in S \mid s \leq t} z_{jst} \geq b_{jt} y_{jt}, \quad j \in J, t \in S \quad (2.8e)$$

$$\sum_{s \in S \mid s \leq t} z_{jst} \leq B_{jt} y_{jt}, \quad j \in J, t \in S \quad (2.8f)$$

$$\sum_{k \in J \mid d_{ik} \leq d_{ij}} x_{iks} \geq y_{jt}, \quad i \in I, j \in J, s, t \in S, s \leq t \quad (2.8g)$$

$$\sum_{j \in J \setminus J_s^0} y_{js} \leq p_s, \quad s \in S \quad (2.8h)$$

$$\sum_{j \in J_s^0} y_{js} \geq |J_s^0| - q_s, \quad \forall s \in S \quad (2.8i)$$

$$x_{ijs} = 0, \quad i \in I, j \in J, s \in S \mid d_{ij} > D_s \quad (2.8j)$$

$$0 \leq x_{ijs} \leq 1, y_{js} \in \{0, 1\}, z_{jst} \geq 0, \quad \forall i \in I, j \in J, s, t \in S, s \leq t \quad (2.8k)$$

La función objetivo (2.8a) considera los costes de asignación de la demanda de todos los niveles atendida por las instalaciones determinadas y los costes fijos correspondientes a la apertura de las instalaciones de todos los niveles en las ubicaciones determinadas. Las restricciones (2.8b) garantizan que todas las demandas de todos los niveles de todos los puntos de demanda sean atendidas. Para cada cliente, instalación y tipo de demanda, las restricciones (2.8c) garantizan que la demanda de un nivel sólo pueda atenderse por una instalación de nivel igual o superior. Estas restricciones impiden realizar asignaciones de puntos de demanda a instalaciones que no se han abierto.

A continuación, se consideran las restricciones que consideran las variables de demanda satisfecha por las instalaciones. Como se ha comentado, z_{jst} es la demanda de nivel s atendida por una instalación de nivel t , con $s \leq t$ ubicada en j . Por lo tanto, las restricciones (2.8d) garantizan que la demanda atendida por una instalación $j \in J$ de nivel s , que viene dada en el término de la derecha de la igualdad, sea igual a la suma de las todas las demandas de nivel s que son atendidas en la ubicación j por instalaciones de nivel s o superior, expresado en la parte izquierda de la igualdad.

Las restricciones (2.8e) y (2.8f) se refieren a las restricciones de capacidad mínima y máxima de las instalaciones de cada nivel de demanda en todas las ubicaciones. Además, si en una ubicación j no se abre una instalación con un nivel de demanda t entonces todas las demandas satisfechas de nivel igual o superior a t de cualquier punto de demanda por esa ubicación j tomarán el valor 0.

Las restricciones (2.8g) corresponden con las restricciones que asignan los puntos de demanda a la instalación más cercana. En este caso, se ha de considerar el nivel de demanda $s \in S$ del cliente $i \in I$ que puede ser atendido por instalaciones del mismo nivel s o superior $t \geq s$ en la ubicación más cercana.

Las ubicaciones $j \in J_s^0$ cuentan con instalaciones abiertas y se puede decidir cerrarlas. En las ubicaciones $j \in J \setminus J_s^0$ no hay instalaciones y se puede decidir abrir una nueva. Teixeira y Antunes (II) introducen las restricciones (2.8h), que limitan el número de nuevas instalaciones atendiendo al parámetro p_s y las restricciones (2.8i), que limitan el número de instalaciones existentes que se pueden cerrar. Estas restricciones no se tendrán en cuenta en los escenarios que se resuelven del modelo jerarquizado en el capítulo 3. Se considera que no hay instalaciones abiertas que se puedan cerrar y que se puede abrir instalaciones en cualquier ubicación.

Las restricciones sobre la distancia máxima de recorrido de los usuarios para acceder a la instalación asignada para cada nivel de demanda se garantizan con el conjunto de restricciones (2.8j). Finalmente, las restricciones (2.8k) corresponde a las restricciones de signo de las variables de decisión.

2.6.2. Variantes del modelo jerarquizado

Al igual que en el modelo (2.3), las restricciones de asignación a la instalación más cercana (2.8g) se pueden sustituir por restricciones de asignación de ruta para cada nivel de demanda:

$$\sum_{k \in P_{ij}} x_{kjs} \geq |P_{ij}|x_{ijs}, \quad i \in I, j \in J, s \in S \quad (2.9)$$

Notar que, en las restricciones de asignación a ruta, las asignaciones de clientes de distintos niveles de demanda son independientes. Sin embargo, en las restricciones (2.8g), la localización de instalaciones de un nivel superior afecta a las asignaciones de niveles inferiores.

En el modelo (2.8) se puede relajar la hipótesis de co-localización e imponer que en cada ubicación $j \in J$ solo se pueda elegir un nivel de demanda. En el nuevo modelo se ha de incluir la siguiente restricción:

$$\sum_{s \in S} y_{js} \leq 1, \quad j \in J \quad (2.10)$$

En este caso, se puede simplificar la definición de las variables z_{jst} e introducir las variables z_{jt} como la demanda total de todos los niveles atendida por la instalación en la ubicación j de nivel t . Las restriccio-

nes (2.8d), (2.8e) y (2.8f) se sustituyen por las siguientes:

$$\sum_{t \in S} z_{jt} = \sum_{s \in S} \sum_{i \in I} u_{is} x_{ijs}, \quad j \in J \quad (2.11a)$$

$$z_{jt} \geq b_{jt} y_{jt}, \quad j \in J, t \in S \quad (2.11b)$$

$$z_{jt} \leq B_{jt} y_{jt}, \quad j \in J, t \in S \quad (2.11c)$$

donde las restricciones (2.11a) definen las variables z_{jt} y las restricciones (2.11b) y (2.11c) garantizan las capacidades mínimas y máximas correspondientes a cada nivel t posible en la ubicación j .

En este TFG se va a considerar un modelo en el que $J_s^0 = \emptyset$, $s \in S$, esto es, no existe ninguna instalación previa y se va a decir dónde ubicar las nuevas instalaciones. En el nuevo modelo se prescinden de las restricciones (2.8i).

Capítulo 3

Aplicación y análisis de los modelos

3.1. Introducción

En este capítulo se describe la experiencia computacional realizada para ilustrar los modelos de localización formulados en el capítulo anterior. Para ello se ha diseñado un ejemplo que se describe en el siguiente apartado. Las funciones para generar los ficheros de datos se han implementado en Python. Los códigos se incluyen en el Anexo.

A continuación, se resuelven todos los modelos. Los problemas se han resuelto en un ordenador Lenovo con el sistema operativo Windows 10 Home 64 bits, con un procesador Intel® Core™ i5-4210U CPU @ 1.70 GHz y una velocidad base de 2.40GHz. En la experimentación se ha utilizado IBM ILOG CPLEX Optimization Studio (versión 20.1.0), un software de optimización que permite resolver problemas de programación lineal, continua y entera, y problemas de programación cuadrática convexa y no convexa con restricciones cuadráticas convexas. Los códigos de todos los modelos se incluyen en el Anexo.

3.2. Descripción del problema

El ejemplo ilustrativo considera 20 nodos que representan tanto los puntos de demanda como las ubicaciones potenciales de las instalaciones. Por tanto, $I = J = \{1, \dots, 20\}$. Las coordenadas de los nodos y las demandas en cada punto, en este ejemplo, se ha utilizado el proporcionado por Texeira and Antunes ([\[1\]](#)). En la Figura [3.1](#) se muestra la ubicación de cada uno de los puntos de demanda, identificados por el número en el centro del círculo, y su demanda, indicada junto al círculo. La Tabla [3.1](#) incluye en la primera columna el número de nodo, en la segunda columna las coordenadas en el plano del nodo y en la tercera columna, la demanda.

Nodo	Coordenadas	Demanda	Nodo	Coordenadas	Demanda
1	(0.8,4)	9	11	(0.1,-4.2)	16
2	(3.46,4.2)	28	12	(-0.74,-1.5)	9
3	(2.14,3.64)	77	13	(-1.1,-0.24)	33
4	(5.66,4.02)	59	14	(-1.28,0.58)	48
5	(6.06,2.7)	86	15	(-0.22,-0.1)	25
6	(7.02,2.56)	6	16	(-0.56,-0.22)	66
7	(5.12,2.18)	52	17	(1.42,1.26)	21
8	(6.68,-0.46)	78	18	(1.48,-1.2)	34
9	(5.88,-0.88)	80	19	(3.96,2.1)	48
10	(3.88,-0.36)	10	20	(-0.2,2.98)	24

Tabla 3.1: Datos del ejemplo ilustrativo con 20 nodos

La demanda total o suma de las demandas de todos los puntos es 809. En el cálculo de los costes

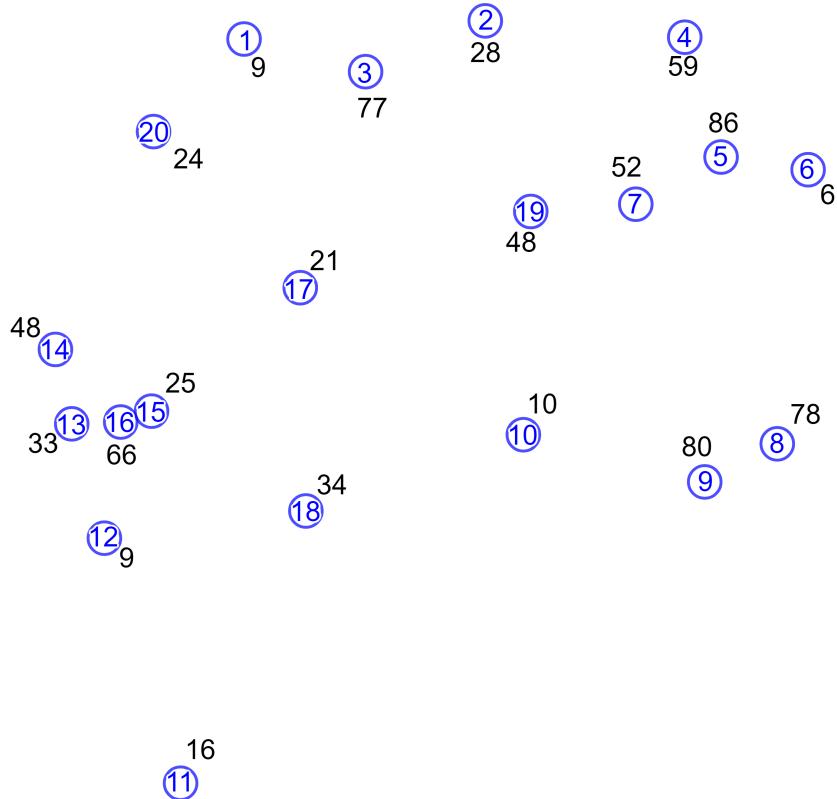


Figura 3.1: Distribución de los nodos y demanda en cada punto

de asignación de un punto de demanda $i \in I$ a una instalación ubicada en $j \in J$ se ha considerado en algunos modelos la distancia Euclídea y en otros modelos, la distancia Manhattan, calculadas a partir de las coordenadas de los puntos. La distancia Euclídea entre dos nodos va de una distancia mínima entre dos nodos de 0.36 a una máxima de 9.92. En el caso de la distancia Mahnhattan, la distancia mínima entre dos nodos es 0.46 y la máxima 13.78.

3.3. Modelos SPLP

El modelo más simple considera costes fijos por la apertura de instalaciones, luego no hay un número predeterminado de instalaciones. Aunque se permite que un punto de demanda sea atendido por varias instalaciones, dado que no hay restricciones de capacidad, el punto de demanda es atendido siempre por la instalación más cercana. Los escenarios resueltos del modelo SPLP (2.1) han considerado exclusivamente la distancia euclídea, no hay restricciones de capacidad en las instalaciones y no consideran niveles de demanda. El único parámetro que se ha modificado son los costes fijos por la apertura y el tratamiento de dichos costes. A continuación, se describen estos escenarios:

SPLP1 Todas las instalaciones son iguales, $f_j = 100$, $j \in J$.

SPLP2 Todas las instalaciones son iguales, $f_j = 200$, $j \in J$.

SPLP3 Hay tres niveles de instalación y por tanto, tres valores de costes fijos:

$$f_j = \begin{cases} 100 & j \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 19, 20\} \\ 250 & j \in \{12, 13, 14, 18\} \\ 500 & j \in \{15, 16, 17\} \end{cases}$$

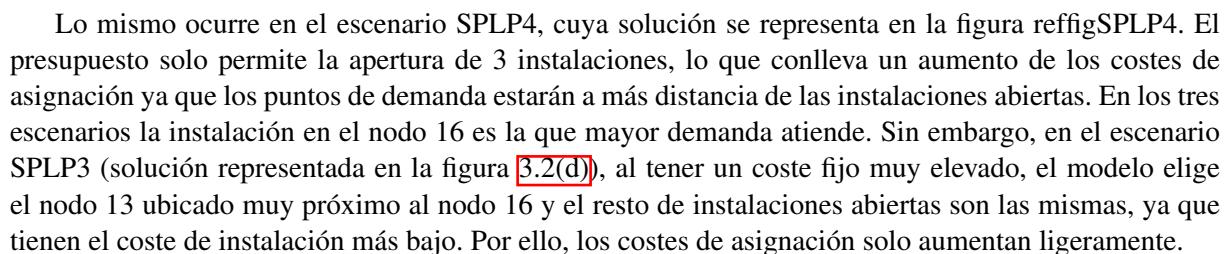
SPLP4 Todas las instalaciones son iguales $f_j = 100$, $j \in J$ y hay un presupuesto de 300 unidades para la ubicación de instalaciones.

En la tabla 3.2 se ha resumido la información de las soluciones óptimas obtenidas. Para cada escenario se incluye en la segunda columna la suma de los costes de asignación de todos los clientes a la instalación correspondiente y en la tercera columna los costes de instalación. En la cuarta columna se indican los nodos en los que se ubican las instalaciones y entre paréntesis, la suma de la demanda de los clientes que atienden. Estos clientes se han incluido en la sexta columna. Finalmente, la última columna indica el tiempo de resolución del problema en segundos.

Resultados de los escenarios del modelo de SPLP					
Escenario	Coste Asignación	Coste Apertura	Instalaciones abiertas(Dem.)	Puntos de demanda asignados	Tiempo de cálculo (sg.)
SPLP1	628.60	500	3 (138) 5(151) 7(100) 9 (168) 16(252)	1,2,3,20 4,5,6 7,19 8,9,10 11,12,13,14,15,16,17,18	0.085
SPLP2	733.48	800	3 (138) 5(251) 9 (168) 16(252)	1,2,3,20 4,5,6,7,19 8,9,10 11,12,13,14,15,16,17,18	0.071
SPLP3	667.24	650	3 (159) 5(151) 7(100) 9 (168) 13(231)	1,2,3,17,20 4,5,6 7,19 8,9,10 11,12,13,14,15,16,18	0.092
SPLP4	1093.52	300	7(356) 9(168) 16(285)	2,3,4,5,6,7,19 8,9,10 1,11,12,13,14,15,16,17,18,20	0.105

Tabla 3.2: Datos de las soluciones óptimas del ejemplo ilustrativo del modelo SPLP

Todos los problemas se han resuelto en menos de 1 segundo. Se observa que al aumentar los costes fijos de apertura en SPLP2 se abre una instalación menos que en el SPLP1, lo que incrementa los costes de asignación de los clientes. Las soluciones óptima se han representado en las figuras 3.2(a) y 3.2(b). En rojo se representan los nodos en los que se ha ubicado una instalación y junto al nodo el valor de la demanda atendida.

Lo mismo ocurre en el escenario SPLP4, cuya solución se representa en la figura  reffigSPLP4. El presupuesto solo permite la apertura de 3 instalaciones, lo que conlleva un aumento de los costes de asignación ya que los puntos de demanda estarán a más distancia de las instalaciones abiertas. En los tres escenarios la instalación en el nodo 16 es la que mayor demanda atiende. Sin embargo, en el escenario SPLP3 (solución representada en la figura 3.2(d)), al tener un coste fijo muy elevado, el modelo elige el nodo 13 ubicado muy próximo al nodo 16 y el resto de instalaciones abiertas son las mismas, ya que tienen el coste de instalación más bajo. Por ello, los costes de asignación solo aumentan ligeramente.

3.4. Modelos de la p-mediana

El modelo de la p-mediana (2.2) es una variante del modelo SPLP (2.1) en el que se introduce el número de instalaciones que se deben abrir. En este caso no se han considerado costes fijos, ni capaci-

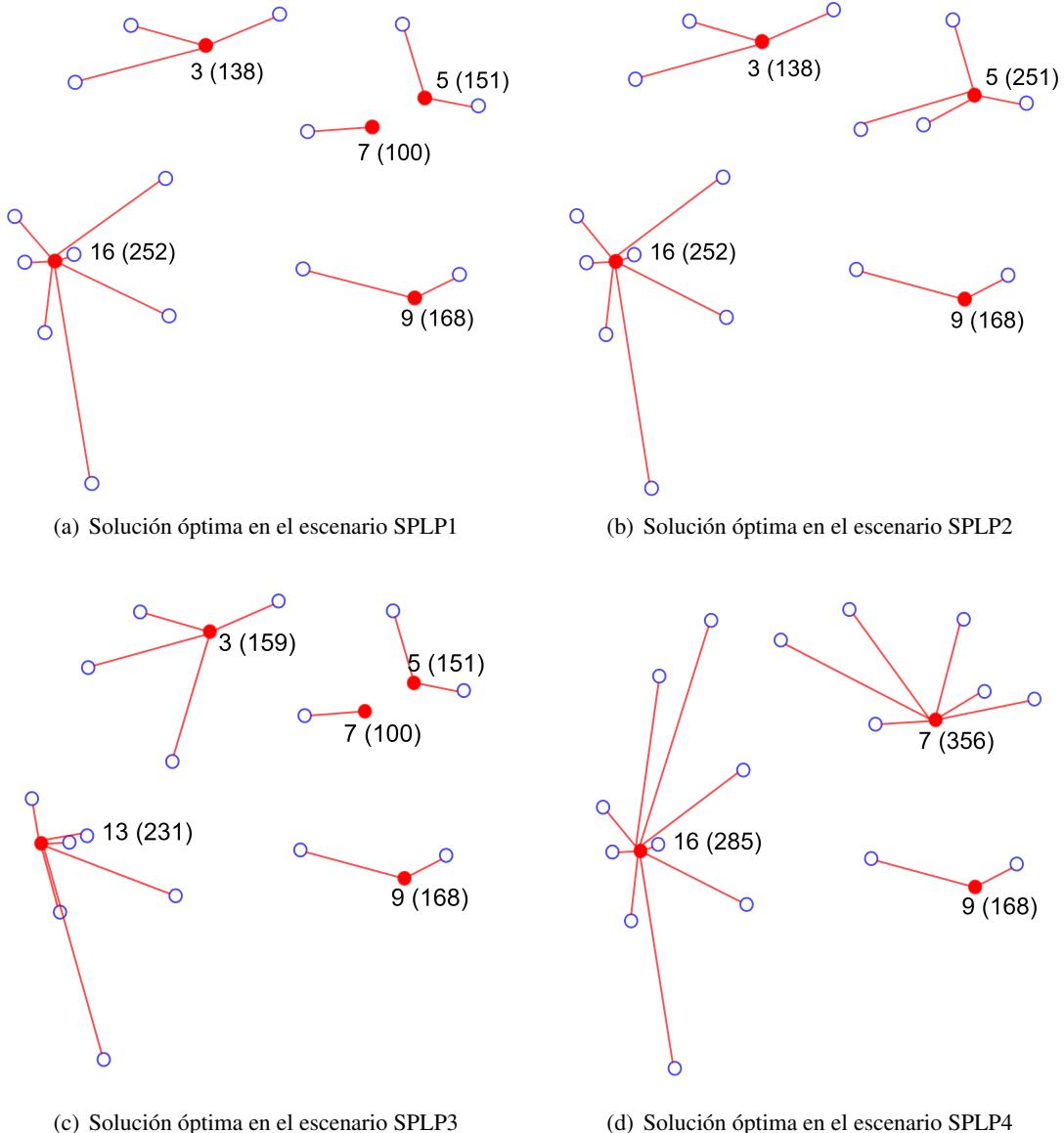


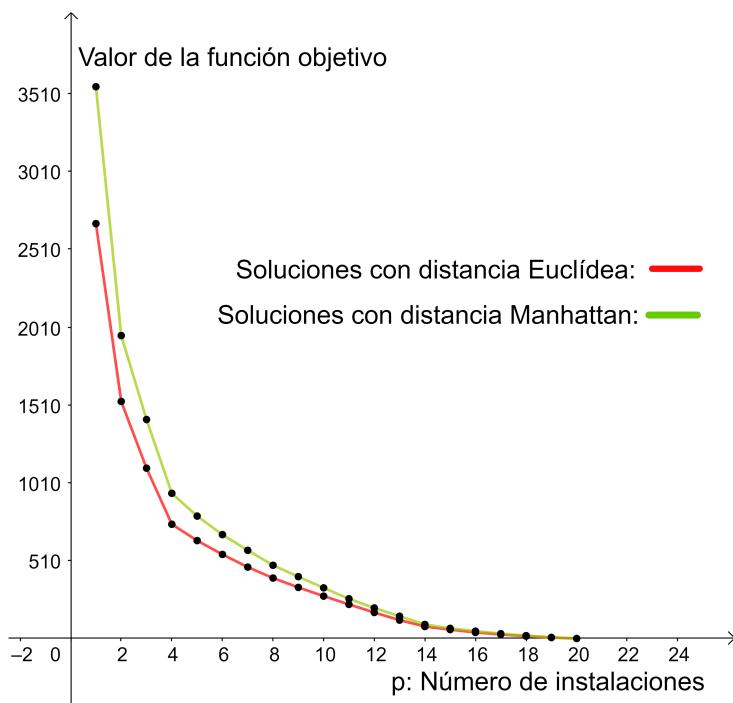
Figura 3.2: Ejemplo ilustrativo del modelo SPLP

dades en las instalaciones ni niveles de demanda. En este modelo se va a modificar el parámetro p y la distancia considerada en los costes. Los escenarios se denotan como $p\text{-med-D}$, donde $p \in \{1, \dots, 20\}$ indica el valor del parámetro p y $D \in \{E, M\}$, siendo E: Euclídea y M: Manhattan. En total se han resuelto 40 escenarios. En la tabla 3.3 se incluye información de las soluciones óptimas de los escenarios con $p \in \{2, 3, 4\}$ y las dos distancias. Todos los tiempos de resolución son inferiores a 1 segundo. Como era de esperar, los valores óptimos de la función objetivo cuando el valor p es el mismo y solo cambia la distancia elegida son mayores con la distancia Manhattan, al ser la distancia euclídea la distancia de la ruta más corta entre dos ubicaciones, luego proporcionan menores costes de asignación.

Notemos que, solo en los escenarios $p = 2$ se eligen diferentes ubicaciones una de las instalaciones abiertas, el nodo 7 para la distancia Euclídea y el nodo 5 para la distancia Manhattan. En ambos se abre la instalación en el nodo 16, pero la asignación de usuarios es diferente. Los escenarios con $p = 3$ la solución óptima coincide y en el caso $p = 4$, aunque las instalaciones abiertas son las mismas, la asignación de los usuarios es diferente, ya que la instalación más cercana es diferente según la distancia utilizada como referencia. En el caso de la distancia Manhattan, el nodo 3 atiende al usuario 17 que en el caso de la distancia Euclídea es atendido por la ubicación en el nodo 16.

Resultados de los escenarios del modelo de la p-mediana				
Escenario	Coste de asignación	Instalaciones abiertas(Dem.)	Puntos de demanda asignados	Tiempo de cálculo (sg.)
2-med-E	1522.06	7(524) 16(285)	2,3,4,5,6,7,8,9,10,19 1,11,12,13,14,15,16,17,18,20	0.112
2-med-M	1945.30	5(514) 16(295)	2,3,4,5,6,7,8,9,19 1,10,11,12,13,14,15,16,17,18,20	0.142
3-med-E	1093.52	7(356) 9(168) 16(285)	2,3,4,5,6,7,19 8,9,10 1,11,12,13,14,15,16,17,18,20	0.124
3-med-M	1405.72	7(356) 9(168) 16(285)	2,3,4,5,6,7,19 8,9,10 1,11,12,13,14,15,16,17,18,20	0.137
4-med-E	733.48	3 (138) 5(251) 9 (168) 16(252)	1,2,3,20 4,5,6,7,19 8,9,10 11,12,13,14,15,16,17,18	0.152
4-med-M	931.95	3 (159) 5(251) 9 (168) 16(231)	1,2,3,17,20 4,5,6,7,19 8,9,10 11,12,13,14,15,16,18	0.158

Tabla 3.3: Datos de las soluciones óptimas del ejemplo ilustrativo del modelo de la p-mediana

Figura 3.3: Valores de la función objetivo según p con las distancias Euclídea y Manhattan

Los valores óptimos de la función objetivo obtenidos en todos los escenarios se representan en la figura 3.4. Se observa, como decrecen los costes totales de asignación hasta 0 al aumentar el número de instalaciones abiertas. Cuando se abren las 20 instalaciones todos los costes de asignación son 0. La distancia total Manhattan siempre es superior a la distancia total Euclídea aunque las diferencias son menores al aumentar p .

3.5. Modelos con capacidad

El modelo con capacidad 2.3 es una variante del modelo SPLP 2.1 en la que se introducen restricciones de capacidad en las instalaciones. En los escenarios de este modelo se han considerado dos tipos de instalación y cada tipo solo se puede instalar en unas ubicaciones dadas. Los datos se incluyen en la tabla 3.6. En los modelos con capacidad se va analizar el efecto de las restricciones de asignación de los

Instalación	Coste fijo	Capacidad mínima	Capacidad máxima	Ubicación
Tipo A	100	20	100	$j \in \{1, 2, 3, 4, 6, 8, 10, 11, 17, 18, 19, 20\}$
Tipo B	200	50	450	$j \in \{5, 7, 9, 12, 13, 14, 15, 16\}$

Tabla 3.4: Datos del modelo con capacidad

usuarios. La suma de las demandas de todos los usuarios es 809 unidades. Esta asignación puede ser a una única instalación o a varias, la asignación a la más cercana y la asignación de ruta. Los escenarios se denotan por CAP-ASI-CRI, donde ASI $\in \{U, M\}$, siendo U: asignación única y M: asignación múltiple; y CRI $\in \{C, RE, RM\}$, siendo C: a la más cercana, RE: asignación de ruta Euclídea con $\epsilon = 1,5$ y RM: asignación de ruta Manhattan con $\epsilon = 1,5$. Además, en el escenario CAP-U-C se ha resuelto una variante de este limitando el presupuesto para la apertura de instalaciones a 600 y por tanto, no se han considerado los costes fijos de apertura en la función objetivo. Se han resuelto un total de ocho escenarios con los datos de capacidad de la tabla 3.6. Toda la información relativa a las soluciones óptimas de los escenarios considerados se han incluido en el Anexo. En la tabla 3.5 se indican las instalaciones abiertas y la asignación de usuarios en las soluciones óptimas.

Un resumen con los aspectos más relevantes que se desean destacar se incluyen en la tabla 3.6. En la primera columna se indica el escenario. El valor óptimo de la función objetivo se muestra separado en dos columnas para distinguir los dos términos del coste, el coste de asignación, en la segunda columna, y el coste de apertura, en la tercera columna. A continuación, se indica en las columnas cuarta y quinta, el número de instalaciones de cada tipo abiertas y en la sexta columna la capacidad máxima de todas. En los modelos con asignación múltiple, en la columna séptima se indica el número de clientes de los 20 que son atendidos por más de una instalación. Finalmente, en la última columna se indica el tiempo de resolución en segundos.

Notar que los tiempos de cálculo son menores en los casos en los que la asignación es única. Los escenarios con menor tiempo de cálculo son CAP-M y CAP-U en los que no se imponen restricciones de asignación de los usuarios. Entre los modelos con asignación única o asignación múltiple, como cabía esperar, siempre proporciona una mejor valor del coste de asignación la asignación múltiple. En el caso de añadir la restricción de asignación de los clientes a la instalación más cercana, se abren un total de siete instalaciones, 4 de tipo A y 3 de tipo B. Por el número de instalaciones abiertas, se obtienen los mayores costes de apertura y los menores costes de asignación de todos los escenarios. Si se restringe el presupuesto a 600, se abren 3 instalaciones de tipo B, reduciendo así los costes de apertura y aumentando los costes de asignación.

En los escenarios CAP-U-C y CAP-M-C proporcionan los mismos resultados porque en este caso al asignar a la instalación más cercana, todos los usuarios se asignan a una única instalación y se precisa la apertura de muchas instalaciones, 7 en total, proporcionando el coste de asignación menor de todos los

Escenario	Instalaciones abiertas	Puntos asignados	Escenario	Instalaciones abiertas	Puntos asignados
CAP-U	3	1,3	CAP-M	3	1,3,20
	5	4-7		5	4-7
	9	8-10		9	8-10
	16	11-18,20		16	11-18,20
	19	2,19		19	2,19
CAP-U-C CAP-M-C	2	2	CAP-U-C<600	7	2-7,19
	3	1,3		9	8-10
	5	4-7		16	1,11-18,20
	9	8-10			
	16	11-16,18			
	19	19			
CAP-U-RE	20	17,20	CAP-M-RE	3	1,2,3,17
	3	3,17		5	4-7
	4	2,4		9	8-10
	6	5,6		16	1,11-18,20
	9	8,9,10		19	2,15,17,18,19
	16	1,11-18,20			
CAP-U-RM	19	7,19	CAP-M-RM	3	1-3
	2	2		5	2,4-7,19
	3	1,3		9	8-10
	5	4-7,19		16	11-16,20
	9	8,9,10		18	11,17,18
	16	11-18,20			

Tabla 3.5: Datos de asignaciones en las soluciones óptimas del modelo con capacidades

Escenario	Coste asignación	Coste Apertura	Instal. tipo A	Instal. tipo B	Capacidad total	Clientes múltiple	CPU (segundos)
CAP-U	668.03	800	2	3	1550		0.285
CAP-M	656.96	800	2	3	1550	1	0.407
CAP-U-C	528	1000	4	3	1750		0.483
CAP-M-C	528	1000	4	3	1750		0.483
CAP-U-C < 600	1093.52	600		3	1350		0.55
CAP-U-RE							
CAP-M-RE	1726.66	600	2	2	1100	12	0.776
CAP-U-RM	900.32	800	2	3	1550		0.473
CAP-M-RM	860.34	800	2	3	1550	2	0.654

Tabla 3.6: Datos de las soluciones óptimas de los escenarios del modelo con capacidades

escenarios. Notemos que, al imponer un presupuesto inferior a 600 y reducir el número de instalaciones abiertas a 3, para que el problema con asignación a la más cercana sea factible abre la instalación ubicada en el nodo 7 (Tabla 3.5) que no es abierta por ninguno de los modelos.

El escenario CAP-U-RE es no factible. Sin embargo, con asignación múltiple el problema es factible. Notemos que, en este escenario CAP-M-RE hay 12 usuarios con asignación múltiple, los costes de asignación son muy elevados y la capacidad máxima de las instalaciones abiertas es la menor de todos los escenarios. Por el contrario, en los escenarios CAP-M y CAP-M-RM, son pocos los usuarios que son atendidos por más de una instalación, uno y dos usuarios, respectivamente.

En los escenarios CAP-U y CAP-M se abren 2 instalaciones de tipo A y 3 de tipo B, al igual que en los escenarios CAP-U-RM y CAP-M-RM. Esto hace que la capacidad total que pueden atender coincida. Los escenarios CAP-U-C y CAP-M-C abren tres instalaciones de tipo B, lo que permite una capacidad máxima de 1350. En el escenario CAP-M-RE, como se ha señalado, la capacidad es menor que en los otros. Se abren 2 instalaciones de tipo A y 2 de tipo B, teniendo una capacidad máxima de 1100.

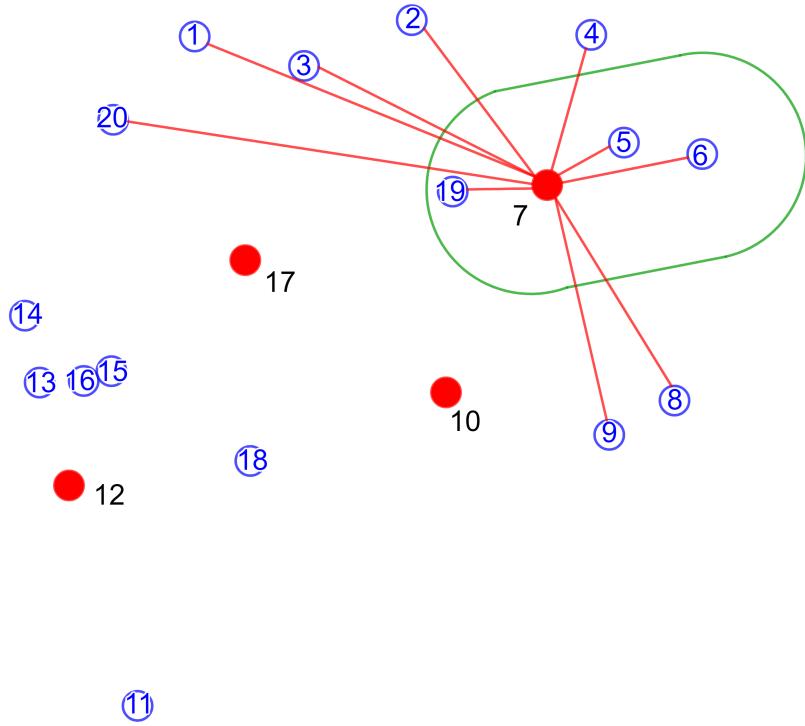


Figura 3.4: Solución óptima de CAP-M-RE y conjunto $P_{6,7}$

En este modelo de capacidad se va a ilustrar la construcción de los conjuntos P_{ij} , $i \in I$, $j \in J$ y las restricciones de asignación a ruta. Se ha elegido la solución óptima al modelo CAP-M-RE, en el que se permite la asignación múltiple y se considera la restricción de asignación a ruta con la distancia Euclídea. En la figura 3.5 se representan con círculos rojos las instalaciones abiertas en las ubicaciones 7, 10, 12 y 17. Debido a las numerosas asignaciones múltiples en este escenario, en la figura se representan solamente las asignaciones a la instalación 7, las asignaciones son las siguientes:

Apertura tipo B en 7: Usuarios de los nodos 1 a 9, 19 y 20.

Apertura tipo A en 10: Usuarios de los nodos 4 a 10 y 19.

Apertura tipo B en 12: Usuarios de los nodos 11 a 16 y 18.

Apertura tipo A en 17: Usuarios de los nodos 1 a 3, 17, 18 y 20.

Por definición, el conjunto $P_{6,7}$ es el subconjunto de puntos que están cerca del camino más corto entre el punto de demanda 6 y la instalación 7, de forma que si el punto de demanda 6 se asigna a la instalación

Card.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
E-0.25	20	289	53	20	13	3			1	1										
E-0.5	18	236	100	24	12	8			1	1										
E-1.5	4	15	48	52	57	74	69	39	23	9	4	6								
M-0.25	20	96	84	58	42	36	18	10	16	8	6	2	2	2						
M-0.5	18	68	78	54	50	40	26	12	16	12	8	10		6		2				
M-1.5	9	16	25	30	56	34	46	34	24	32	18	12	10	8	4	8	2	12	16	4

Tabla 3.7: Cardinalidad de los conjuntos P_{ij} , $i \in I$, $j \in J$ según distancia y ε

7, todos los clientes situados a menos de $\varepsilon = 1,5$ km de la ruta será asignados a la instalación 7. En la figura 3.5, en verde, se delimita el área de proximidad a la ruta del nodo 6 al nodo 7 y que incluye a 4 clientes, $P_{6,7} = \{5, 6, 7, 19\}$. Notemos que, todos ellos están asignados a la instalación 7. En ese caso, se permite asignación múltiple, así los clientes 5, 6, 7 y 19 se dirigen a más de una instalación ya que la instalación 10 también los atiende.

El cálculo de los conjuntos P_{ij} , $i \in I$, $j \in J$ se realiza una sola vez, antes de resolver el problema y su cardinalidad depende del valor ε elegido. De forma ilustrativa, en la siguiente matriz, el elemento (i, j) indica el cardinal del correspondiente conjunto P_{ij} en el escenario CAP-M-RE con $\varepsilon=1.5$:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	3	4	4	5	8	6	6	5	5	5	4	7	8	7	8	8	4	4	5	3
2	5	2	5	4	6	4	5	5	4	3	3	6	7	6	7	7	3	3	4	4
3	9	3	3	5	7	6	6	6	5	4	4	7	8	7	8	8	4	4	5	7
4	7	5	6	2	4	3	4	4	4	3	3	6	7	6	7	7	3	3	4	4
5	7	8	7	6	4	12	8	6	6	5	5	8	9	8	9	9	5	5	11	8
6	6	5	6	6	6	2	10	4	4	3	3	6	7	6	7	7	3	3	11	7
7	9	6	9	4	10	4	3	5	5	4	4	7	8	7	8	8	9	4	3	8
8	5	4	6	6	6	5	2	12	6	3	6	7	8	8	8	6	6	9	7	
9	6	5	6	7	6	6	7	7	2	10	3	6	7	8	8	8	4	7	5	6
10	4	5	7	3	6	3	7	3	3	1	2	5	7	7	7	7	8	12	9	5
11	6	3	4	3	5	3	4	3	3	2	1	5	6	6	9	7	7	3	5	
12	9	6	8	6	8	6	7	6	6	5	5	4	9	5	5	5	7	5	6	8
13	9	7	8	7	9	7	8	7	7	6	6	5	5	11	8	6	8	6	7	7
14	8	6	7	6	8	6	7	6	6	5	5	5	6	4	5	7	5	5	6	8
15	10	7	8	7	9	7	8	7	7	6	6	5	8	10	5	5	9	6	7	9
16	9	7	9	7	9	7	8	7	7	6	6	5	5	11	5	5	9	6	7	8
17	4	5	6	3	5	3	4	4	3	2	2	5	6	12	6	6	1	2	3	6
18	7	4	4	3	5	3	4	3	3	2	2	12	10	9	10	12	2	1	8	6
19	10	4	10	6	5	4	3	4	4	3	3	6	7	6	7	7	9	3	2	7
20	3	4	4	4	6	4	5	4	4	3	3	6	7	6	7	7	3	3	4	2

Con la distancia Euclídea, el máximo cardinal es 12 y son 6 los conjuntos P_{ij} que tienen este cardinal. El cardinal mínimo es 1 y son 20 los conjuntos que tienen este cardinal. Esta matriz es simétrica ya que la ruta que va de i a j es la misma que va de j a i , por lo que las instalaciones cercanas a dicha ruta coinciden. En la tabla 3.7 se introduce un resumen sobre la cardinalidad de los conjuntos según la distancia y el valor de ε . En cada columna se muestra el número de conjuntos con el cardinal indicado en la primera fila. Segundo aumenta el valor de ε , en ambas distancias, aumenta la frecuencia de los valores de cardinal más altos. El área determinada por la distancia Manhattan, para un mismo valor de ε es mayor que el área determinada por la distancia Euclídea, por ello valores de cardinal hasta 20 solo se observan con la distancia Manhattan.

3.6. Modelo jerarquizado

Este modelo es una variante del modelo con capacidad en el que se incluyen tipos de instalaciones y niveles de demanda. En el ejemplo ilustrativo se van a considerar dos niveles de demanda, $S = \{1, 2\}$ y por tanto, en cada ubicación se pueden establecer instalaciones de dos niveles. Además, las instalaciones son de diferente tamaño y el valor de los parámetros para los cuatro tipos considerados se incluyen en la tabla 3.8. El tipo de ubicación en cada uno de los 20 nodos se elige al azar y según el tipo asignado se elige al azar una demanda para cada nivel según la distribución uniforme discreta indicada en las dos últimas columnas. La generación de estos valores se realiza con un programa implementado en Python. El resultado se muestra en la tabla 3.9. El valor de la demanda total de nivel 1 es 1058 y de nivel 2 es 425.

Tipo de ubicación	Coste fijo		Cap. mínima		Cap. máxima		Demanda	
	Nivel 1	Nivel 2	Nivel 1	Nivel 2	Nivel 1	Nivel 2	Nivel 1	Nivel 2
A	100	500	20	50	60	150	{10,9,11}	{1,0,2}
B	250	1250	50	70	150	210	{25,23,28}	{8,7,9}
C	750	3750	150	200	450	600	{75,68,83}	{30,27,33}
D	1000	5000	200	300	600	900	{100,90,110}	{45,41,50}

Tabla 3.8: Datos de los cuatro tipos de ubicación para cada nivel de demanda

Nodo	1	2	3	4	5	6	7	8	9	10
Tipo	A	C	A	B	B	A	D	C	C	D
Demanda nivel 1	9	75	9	23	25	11	110	83	75	110
Demanda nivel 2	1	30	1	8	7	1	45	33	27	41

Nodo	11	12	13	14	15	16	17	18	19	20
Tipo	A	A	A	A	D	D	B	C	D	D
Demanda nivel 1	10	5	11	11	100	90	23	68	110	100
Demanda nivel 2	1	9	1	1	45	45	8	30	41	50

Tabla 3.9: Datos del ejemplo ilustrativo

Con este modelo se plantean los escenarios atendiendo a si se permite o no la co-localización y si se añaden restricciones al número de instalaciones de cada nivel. En todos los escenarios se realiza la asignación única a la instalación más cercana. Los escenarios se denotan por JER-z-(n1,n2), donde $z \in \{0, 1\}$ toma el valor 1 si se permite la co-localización y 0, en otro caso; (n1,n2) toma el valor (-,-) si no hay limitaciones y si es no nulo, indica el número máximo de instalaciones de cada uno de los dos niveles. En todos los escenarios se ha tomado una distancia máxima de los usuarios a las instalaciones de ambos niveles igual a 4. En el caso de los escenarios con limitación en el número de instalaciones los problemas no son factibles. En el ejemplo, la distancia Euclídea entre dos nodos mínima es 0.54 y la máxima 9. En los escenarios con limitación (3,2) se han resuelto con una distancia máxima de 5 para garantizar la factibilidad y en el caso de los escenarios con limitación (2,1), la distancia máxima ha sido 6.

Toda la información relativa a las soluciones óptimas de los escenarios considerados se han incluido en el Anexo. Un resumen con los aspectos más relevantes que se desean destacar se incluyen en las tablas 3.10 y 3.11. En la primera columna se muestra el escenario y los costes de asignación y de apertura de instalaciones en la segunda y tercera columna. En las siguientes tres columnas se indica el número de ubicaciones en las que se abren solo instalaciones de nivel 1, solo de nivel 2, y de ambos niveles, respectivamente. En la séptima columna la capacidad máxima de las instalaciones abiertas de nivel 1 y

en la octava columna, de nivel 2. En la novena columna se indica la demanda total satisfecha de nivel 1 por una instalación de nivel 2. Finalmente, el tiempo requerido para el cálculo se ha incluido en la tabla 3.11.

Escenario	Coste asig.	Coste Apert.	Inst. tipo 1	Inst. tipo 2	Inst. 1 y 2	Cap. max. nivel 1	Cap. max. nivel 2	Demanda 1 atendida por 2
JER-0-(-,-)	2475.36	4600	2	1	2	1410	1260	75
JER-1-(-,-)	1952.66	7500	4	4		1350	1050	327
JER-0-(2,1)	3116.15	6750	2	1		1050	900	268
JER-1-(2,1)	3116.15	6750	2	1		1050	900	268
JER-0-(3,2)	2740.92	6750	2	2		750	1050	596
JER-1-(3,2)	2740.92	6750	2	2		750	1050	596
JER-0-(4,3)	2475.36	4600	2	1	2	1410	1260	75
JER-1-(4,3)	2499.12	7750	1	3		600	1260	778

Tabla 3.10: Datos de las soluciones óptimas de los escenarios del modelo jerarquizado

Escenario	CPU (segundos)
JER-0-(-,-)	1.314
JER-1-(-,-)	2.97
JER-0-(2,1)	2.223
JER-1-(2,1)	2.628
JER-0-(3,2)	1.5
JER-1-(3,2)	1.731
JER-0-(4,3)	1.358
JER-1-(4,3)	1.895

Tabla 3.11: Tiempos de resolución de los escenarios del modelo con jerarquizado

El menor coste de asignación lo proporciona el modelo con co-localización sin límite de apertura de instalaciones, si bien es el modelo con mayor coste de apertura, ya que abre 4 plantas de

Se observa, que en los modelos con co-localización sin límite de apertura de instalaciones los costes de asignación son inferiores al ser modelos menos restringidos. Aumentan también los costes totales de apertura por instalar un mayor número de instalaciones. En los escenarios JER-0-(4,3) y JER-1-(4,3) coinciden las soluciones con los escenarios en los que no hay límite de apertura de instalaciones.

Cuando se restringen más el número de instalaciones que se pueden abrir, las soluciones de los escenarios con co-localización y sin co-localización coinciden.

Notemos que, los tiempos de cálculo son muy superiores a los escenarios resueltos de otros modelos. Esto se debe a que en el modelo jerarquizado, al introducir los niveles de demanda se aumenta la complejidad, aumentando así el coste computacional del cálculo de las soluciones.

Escenario	Instalaciones abiertas(nivel)	Puntos de demanda asignados de nivel 1	Puntos de demanda asignados de nivel 2
JER-0-(-,-)	1(1,2)	1,3,20	1,2,3,17,20
	5(1,2)	4-6	4-10,19
	7(1)	2,7-10,19	
	12(2)	11,12	11-16,18
	15(1)	13-18	
JER-1-(-,-)	2	2,19	
	3(2)	1,3	1-3,17,19,20
	4(1)	4,7	
	6(2)	5,6	4-7
	9(2)	8-10	8-10
	12(2)	11,12	11-16,18
	15(1)	13-16,18	
JER-0-(2,1)	17(1)	17,20	
	2(1)	1-7,19	
	10(2)	8-10	1-20
JER-1-(2,1)	15(1)	11-18,20	
	2(1)	1-7,19	
	10(2)	8-10	1-20
	15(1)	11-18,20	
JER-0-(3,2)	4(1)	4,5,6	
	12(2)	11,12	11-16,18
	15(1)	13-18,20	
	19(2)	1-3,7-10,19	1-10,17-20
JER-1-(3,2)	4(1)	4,5,6	
	12(2)	11,12	11-16,18
	15(1)	13-18,20	
	19(2)	1-3,7-10,19	1-10,17-20
JER-0-(4,3)	1(1,2)	1,3,20	1,2,3,17,20
	5(1,2)	4-6	4-10,19
	7(1)	2,7-10,19	
	12(2)	11,12	11-16,18
	15(1)	13-18	
JER-1-(4,3)	17(2)	1,3,17,20	1,3,17,20
	7(2)	2,4-10,19	
	12(2)	11,12	11-16,18
	15(1)	1,13-16,18	

Tabla 3.12: Datos de asignaciones en las soluciones óptimas del modelo jerarquizado

Bibliografía

- [1] JOÃO C. TEIXEIRA AND ANTÓNIO P. ANTUNES, A hierarchical location model for public facility planning, (2006).
- [2] VLADIMIR MARIANOV AND DANIEL SERRA, Location Problems in the Public Sector, (2002).
- [3] C.S. REVELLE AND H.A. EISELT, Location analysis: A synthesis and survey, (2003).
- [4] H. A. EISELT, VLADIMIR MARIANOV AND JOYENDU BHADURY, Applications of Location Analysis. (2015)
- [5] MIQUEL CARRERAS AND DANIEL SERRA , On optimal location with threshold requirements. (1999)

Anexo

June 26, 2022

1 Código y datos del modelo SPLP en IBM ILOG CPLEX Optimization Studio

```
[ ]: /*****
 * OPL 22.1.0.0 Model
 * Author: Usuario
 * Creation Date: 7 jun. 2022 at 12:29:14
 *****/
// MODELO SPLP

// Defino las variables fijas
int n = ...; //número de cliente
int m = ...; //número de ubicaciones

range cliente=1..n;
range ubicacion=1..m;

float u[cliente]=...;

float f[ubicacion]=...;

float distancia[cliente][ubicacion] = ... ;

// Defino variables de decision
dvar boolean x[cliente][ubicacion];
dvar boolean y[ubicacion];

minimize sum (i in cliente,j in ubicacion) (distancia[i][j]*u[i]*x[i][j]) + sum(j in ubicacion) f[j]*y[j];
subject to {
//cada cliente es atendido por una y solo una instalación
forall (i in cliente) sum(j in ubicacion)
    x[i][j] == 1;
//un cliente es atendido por una instalacion que esta abierta
```

```

forall (i in cliente, j in ubicacion)
    x[i][j] <= y[j];

}

main {
    var f = new IloOplOutputFile("solucionSPLP1.txt");
    thisOplModel.generate();
    var time0 = new Date();
    var OK = cplex.solve();
    var time1 = new Date();

    var tTotal = (time1.getTime()-time0.getTime())/1000;

    if (OK) {
        f.writeln("Instalaciones abiertas");
        for(var j in thisOplModel.ubicacion){
            if(thisOplModel.y[j] != 0){
                f.writeln("Abro instalacion ",j);
            }
        }
        f.writeln("Asignacion de clientes");
        for(var j in thisOplModel.ubicacion){
            for(var i in thisOplModel.cliente){
                if(thisOplModel.x[i][j] != 0){
                    f.writeln("a la instalacion ",j," van",
                     "los clientes ",i);
                }
            }
        }
        f.writeln("Tiempo: ",tTotal,"segundos. Objetivo: "+cplex.
        →getObjValue( ));
    }
    f.close();
}

```

```

[ ]: ****
* OPL 22.1.0.0 Data
* Author: Usuario
* Creation Date: 7 jun. 2022 at 12:29:14
****

//Numero de nodos
n=20;
m=20;

```

```

//DATOS SPLP

//Demanda en los puntos
u=[9,28,77,59,86,6,52,78,80,10,16,9,33,48,25,66,21,34,48,24];

// Costes fijos

f=[100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100];
→100, 100, 100, 100, 100];

//SE CAMBIA f CON LOS DATOS DE CADA ESCENARIO

//Distancia entre los nodos
distancia=[[0, 2.6675082 , 1.38751577, 4.86004115, 5.41826541,
6.38451251, 4.68772866, 7.3801084 , 7.04420329, 5.33816448,
8.22982381, 5.71153219, 4.6462458 , 4.00284899, 4.22497337,
4.43312125, 2.80927037, 5.24453277, 3.68893535, 1.42702628],
[2.6675082 , 0.          , 1.43387587, 2.20735135, 3.0016662 ,
3.91959182, 2.61457454, 5.664274 , 5.62697077, 4.57930126,
9.04707688, 7.08025423, 6.36452669, 5.96422669, 5.65971731,
5.97333441, 3.57843541, 5.75086811, 2.15916743, 3.85608143],
[1.38751577, 1.43387587, 0.          , 3.54045195, 4.03112887,
4.99807963, 3.31843337, 6.11731967, 5.86668561, 4.36206373,
8.10106166, 5.89185879, 5.05489861, 4.58911756, 4.42235232,
4.70943776, 2.48652368, 4.8845229 , 2.38564121, 2.42937111],
[4.86004115, 2.20735135, 3.54045195, 0.          , 1.37927517,
1.99529446, 1.91760267, 4.59464906, 4.90493629, 4.72787479,
9.92380975, 8.45165073, 7.99031914, 7.74578595, 7.1797493 ,
7.52602976, 5.05916989, 6.68611128, 2.56312388, 5.949602 ],
[5.41826541, 3.0016662 , 4.03112887, 1.37927517, 0.          ,
0.97015463, 1.07424392, 3.22024844, 3.58452228, 3.75712656,
9.11765321, 7.99249648, 7.74010336, 7.64002618, 6.87592903,
7.23355542, 4.85831246, 6.01399069, 2.18210999, 6.26426085],
[6.38451251, 3.91959182, 4.99807963, 1.99529446, 0.97015463,
0.          , 1.93762742, 3.03907881, 3.62397572, 4.28788992,
9.67388236, 8.75792213, 8.58920252, 8.53290103, 7.71318352,
8.07183275, 5.74891294, 6.69380639, 3.09240424, 7.23020913],
[4.68772866, 2.61457454, 3.31843337, 1.91760267, 1.07424392,
1.93762742, 0.          , 3.06646376, 3.15296686, 2.82651729,
8.11817714, 6.91968207, 6.67418909, 6.596969 , 5.80637581,
6.16438837, 3.81266311, 4.96582762, 1.1607601 , 5.37783637],
[7.3801084 , 5.664274 , 6.11731967, 4.59464906, 3.22024844,
3.03907881, 3.06646376, 0.          , 0.90354856, 2.80178515,
7.56861942, 7.49252961, 7.78310992, 8.02765221, 6.90938492,
7.24197791, 5.53407626, 5.25040989, 3.73378146, 7.69028504],

```

[7.04420329, 5.62697077, 5.86668561, 4.90493629, 3.58452228,
 3.62397572, 3.15296686, 0.90354856, 0. , 2.06649462,
 6.66564325, 6.64896985, 7.00927956, 7.30733878, 6.14966666,
 6.47174196, 4.94683737, 4.40962629, 3.54388544, 7.20011694],
 [5.33816448, 4.57930126, 4.36206373, 4.72787479, 3.75712656,
 4.28788992, 2.82651729, 2.80178515, 2.06649462, 0. ,
 5.3883207 , 4.75857121, 4.98144557, 5.24492135, 4.10823563,
 4.44020765, 2.94550505, 2.54086678, 2.46136629, 5.27121276],
 [8.22982381, 9.04707688, 8.10106166, 9.92380975, 9.11765321,
 9.67388236, 8.11817714, 7.56861942, 6.66564325, 5.3883207 ,
 0. , 2.8276492 , 4.13782552, 4.97521859, 4.11246884,
 4.03402578, 5.61729472, 3.30301741, 7.38952258, 7.18618146],
 [5.71153219, 7.08025423, 5.89185879, 8.45165073, 7.99249648,
 8.75792213, 6.91968207, 7.49252961, 6.64896985, 4.75857121,
 2.8276492 , 0. , 1.31041978, 2.14895323, 1.49345238,
 1.29287432, 3.50473965, 2.24216057, 5.92189193, 4.51266706],
 [4.6462458 , 6.36452669, 5.05489861, 7.99031914, 7.74010336,
 8.58920252, 6.67418909, 7.78310992, 7.00927956, 4.98144557,
 4.13782552, 1.31041978, 0. , 0.83952367, 0.89106678,
 0.54236888, 2.93264386, 2.75469127, 5.57668755, 3.34395036],
 [4.00284899, 5.96422669, 4.58911756, 7.74578595, 7.64002618,
 8.53290103, 6.596969 , 8.02765221, 7.30733878, 5.24492135,
 4.97521859, 2.14895323, 0.83952367, 0. , 1.25936492,
 1.07762888, 2.7843132 , 3.28588557, 5.45792671, 2.63262683],
 [4.22497337, 5.65971731, 4.42235232, 7.1797493 , 6.87592903,
 7.71318352, 5.80637581, 6.90938492, 6.14966666, 4.10823563,
 4.11246884, 1.49345238, 0.89106678, 1.25936492, 0. ,
 0.35866976, 2.13053984, 2.0265251 , 4.72537025, 3.08007857],
 [4.43312125, 5.97333441, 4.70943776, 7.52602976, 7.23355542,
 8.07183275, 6.16438837, 7.24197791, 6.47174196, 4.44020765,
 4.03402578, 1.29287432, 0.54236888, 1.07762888, 0.35866976,
 0. , 2.47040159, 2.2631836 , 5.08062988, 3.22018633],
 [2.80927037, 3.57843541, 2.48652368, 5.05916989, 4.85831246,
 5.74891294, 3.81266311, 5.53407626, 4.94683737, 2.94550505,
 5.61729472, 3.50473965, 2.93264386, 2.7843132 , 2.13053984,
 2.47040159, 0. , 2.46078118, 2.67719331, 2.36142415],
 [5.24453277, 5.75086811, 4.8845229 , 6.68611128, 6.01399069,
 6.69380639, 4.96582762, 5.25040989, 4.40962629, 2.54086678,
 3.30301741, 2.24216057, 2.75469127, 3.28588557, 2.0265251 ,
 2.2631836 , 2.46078118, 0. , 4.12800194, 4.50497503],
 [3.68893535, 2.15916743, 2.38564121, 2.56312388, 2.18210999,
 3.09240424, 1.1607601 , 3.73378146, 3.54388544, 2.46136629,
 7.38952258, 5.92189193, 5.57668755, 5.45792671, 4.72537025,
 5.08062988, 2.67719331, 4.12800194, 0. , 4.25205833],
 [1.42702628, 3.85608143, 2.42937111, 5.949602 , 6.26426085,
 7.23020913, 5.37783637, 7.69028504, 7.20011694, 5.27121276,
 7.18618146, 4.51266706, 3.34395036, 2.63262683, 3.08007857,

```
3.22018633, 2.36142415, 4.50497503, 4.25205833, 0.           ]];
```

2 Código y datos del modelo de la p -mediana en IBM ILOG CPLEX Optimization Studio

```
[ ]: ****  
* OPL 22.1.0.0 Model  
* Author: Usuario  
* Creation Date: 10 jun. 2022 at 8:33:00  
*****  
  
// MODELO P-MEDIANA CON DISTANCIA EUCLIDEA  
  
// Defino las variables fijas  
int n = ...; //número de cliente  
int m = ...; //número de ubicaciones  
int p=...; //número de instalaciones a ubicar  
  
range cliente=1..n;  
range ubicacion=1..m;  
  
float u[cliente]=...;  
  
float distancia[cliente][ubicacion] = ... ;  
  
// Defino variables de decisión  
dvar boolean x[cliente][ubicacion];  
dvar boolean y[ubicacion];  
  
minimize sum (i in cliente,j in ubicacion) distancia[i][j]*u[i]*x[i][j];  
subject to {  
//cada cliente es atendido por una y solo una instalación  
forall (i in cliente) sum(j in ubicacion)  
    x[i][j] == 1;  
//un cliente es atendido por una instalación que está abierta  
forall (i in cliente, j in ubicacion)  
    x[i][j] <= y[j];  
  
//se abren justamente p instalaciones  
sum(j in ubicacion) y[j] == p ;  
}
```

```

main {
    var f = new IloOplOutputFile("solucionpmed.txt");
    thisOplModel.generate();
    var time0 = new Date();
    var OK = cplex.solve();
    var time1 = new Date();

    var tTotal = (time1.getTime()-time0.getTime())/1000;

    if (OK) {
        f.writeln("Instalaciones abiertas");
        for(var j in thisOplModel.ubicacion){
            if(thisOplModel.y[j] != 0){
                f.writeln("Abro instalacion ",j);
            }
        }
        f.writeln("Asignacion de clientes");
        for(var j in thisOplModel.ubicacion){
            for(var i in thisOplModel.cliente){
                if(thisOplModel.x[i][j] != 0){
                    f.writeln("a la instalacion ",j," van",
                     "los clientes ",i);
                }
            }
        }
        f.writeln("Tiempo: ",tTotal,"segundos. Objetivo: "+cplex.
        →getObjValue());
    }
    f.close();
}

```

```

[ ]: ****
* OPL 22.1.0.0 Data
* Author: pardo
* Creation Date: 10 jun. 2022 at 8:33:00
*****

```

```

//DATOS DE LOS MODELOS P-MEDIANA CON DISTANCIA EUCLIDEA

//Numero de nodos
n=20;
m=20;
p=2;
//Se cambia p para cada escenario

```

```

//Demanda en los puntos
u=[9,28,77,59,86,6,52,78,80,10,16,9,33,48,25,66,21,34,48,24];

//Distancia entre los nodos
distancia=[[0, 2.6675082 , 1.38751577, 4.86004115, 5.41826541,
6.38451251, 4.68772866, 7.3801084 , 7.04420329, 5.33816448,
8.22982381, 5.71153219, 4.6462458 , 4.00284899, 4.22497337,
4.43312125, 2.80927037, 5.24453277, 3.68893535, 1.42702628],
[2.6675082 , 0. , 1.43387587, 2.20735135, 3.0016662 ,
3.91959182, 2.61457454, 5.664274 , 5.62697077, 4.57930126,
9.04707688, 7.08025423, 6.36452669, 5.96422669, 5.65971731,
5.97333441, 3.57843541, 5.75086811, 2.15916743, 3.85608143],
[1.38751577, 1.43387587, 0. , 3.54045195, 4.03112887,
4.99807963, 3.31843337, 6.11731967, 5.86668561, 4.36206373,
8.10106166, 5.89185879, 5.05489861, 4.58911756, 4.42235232,
4.70943776, 2.48652368, 4.8845229 , 2.38564121, 2.42937111],
[4.86004115, 2.20735135, 3.54045195, 0. , 1.37927517,
1.99529446, 1.91760267, 4.59464906, 4.90493629, 4.72787479,
9.92380975, 8.45165073, 7.99031914, 7.74578595, 7.1797493 ,
7.52602976, 5.05916989, 6.68611128, 2.56312388, 5.949602 ],
[5.41826541, 3.0016662 , 4.03112887, 1.37927517, 0. ,
0.97015463, 1.07424392, 3.22024844, 3.58452228, 3.75712656,
9.11765321, 7.99249648, 7.74010336, 7.64002618, 6.87592903,
7.23355542, 4.85831246, 6.01399069, 2.18210999, 6.26426085],
[6.38451251, 3.91959182, 4.99807963, 1.99529446, 0.97015463,
0. , 1.93762742, 3.03907881, 3.62397572, 4.28788992,
9.67388236, 8.75792213, 8.58920252, 8.53290103, 7.71318352,
8.07183275, 5.74891294, 6.69380639, 3.09240424, 7.23020913],
[4.68772866, 2.61457454, 3.31843337, 1.91760267, 1.07424392,
1.93762742, 0. , 3.06646376, 3.15296686, 2.82651729,
8.11817714, 6.91968207, 6.67418909, 6.596969 , 5.80637581,
6.16438837, 3.81266311, 4.96582762, 1.1607601 , 5.37783637],
[7.3801084 , 5.664274 , 6.11731967, 4.59464906, 3.22024844,
3.03907881, 3.06646376, 0. , 0.90354856, 2.80178515,
7.56861942, 7.49252961, 7.78310992, 8.02765221, 6.90938492,
7.24197791, 5.53407626, 5.25040989, 3.73378146, 7.69028504],
[7.04420329, 5.62697077, 5.86668561, 4.90493629, 3.58452228,
3.62397572, 3.15296686, 0.90354856, 0. , 2.06649462,
6.66564325, 6.64896985, 7.00927956, 7.30733878, 6.14966666,
6.47174196, 4.94683737, 4.40962629, 3.54388544, 7.20011694],
[5.33816448, 4.57930126, 4.36206373, 4.72787479, 3.75712656,
4.28788992, 2.82651729, 2.80178515, 2.06649462, 0. ,
5.3883207 , 4.75857121, 4.98144557, 5.24492135, 4.10823563,
4.44020765, 2.94550505, 2.54086678, 2.46136629, 5.27121276],
[8.22982381, 9.04707688, 8.10106166, 9.92380975, 9.11765321,
9.67388236, 8.11817714, 7.56861942, 6.66564325, 5.3883207 ,
5.3883207 , 4.75857121, 4.98144557, 5.24492135, 4.10823563,
4.44020765, 2.94550505, 2.54086678, 2.46136629, 5.27121276],
[8.22982381, 9.04707688, 8.10106166, 9.92380975, 9.11765321,
9.67388236, 8.11817714, 7.56861942, 6.66564325, 5.3883207 ,
5.3883207 , 4.75857121, 4.98144557, 5.24492135, 4.10823563,
4.44020765, 2.94550505, 2.54086678, 2.46136629, 5.27121276]
]
```

```

0.          , 2.8276492 , 4.13782552, 4.97521859, 4.11246884,
4.03402578, 5.61729472, 3.30301741, 7.38952258, 7.18618146], 
[5.71153219, 7.08025423, 5.89185879, 8.45165073, 7.99249648,
8.75792213, 6.91968207, 7.49252961, 6.64896985, 4.75857121,
2.8276492 , 0.          , 1.31041978, 2.14895323, 1.49345238,
1.29287432, 3.50473965, 2.24216057, 5.92189193, 4.51266706], 
[4.6462458 , 6.36452669, 5.05489861, 7.99031914, 7.74010336,
8.58920252, 6.67418909, 7.78310992, 7.00927956, 4.98144557,
4.13782552, 1.31041978, 0.          , 0.83952367, 0.89106678,
0.54236888, 2.93264386, 2.75469127, 5.57668755, 3.34395036], 
[4.00284899, 5.96422669, 4.58911756, 7.74578595, 7.64002618,
8.53290103, 6.596969 , 8.02765221, 7.30733878, 5.24492135,
4.97521859, 2.14895323, 0.83952367, 0.          , 1.25936492,
1.07762888, 2.7843132 , 3.28588557, 5.45792671, 2.63262683], 
[4.22497337, 5.65971731, 4.42235232, 7.1797493 , 6.87592903,
7.71318352, 5.80637581, 6.90938492, 6.14966666, 4.10823563,
4.11246884, 1.49345238, 0.89106678, 1.25936492, 0.          ,
0.35866976, 2.13053984, 2.0265251 , 4.72537025, 3.08007857], 
[4.43312125, 5.97333441, 4.70943776, 7.52602976, 7.23355542,
8.07183275, 6.16438837, 7.24197791, 6.47174196, 4.44020765,
4.03402578, 1.29287432, 0.54236888, 1.07762888, 0.35866976,
0.          , 2.47040159, 2.2631836 , 5.08062988, 3.22018633], 
[2.80927037, 3.57843541, 2.48652368, 5.05916989, 4.85831246,
5.74891294, 3.81266311, 5.53407626, 4.94683737, 2.94550505,
5.61729472, 3.50473965, 2.93264386, 2.7843132 , 2.13053984,
2.47040159, 0.          , 2.46078118, 2.67719331, 2.36142415], 
[5.24453277, 5.75086811, 4.8845229 , 6.68611128, 6.01399069,
6.69380639, 4.96582762, 5.25040989, 4.40962629, 2.54086678,
3.30301741, 2.24216057, 2.75469127, 3.28588557, 2.0265251 ,
2.2631836 , 2.46078118, 0.          , 4.12800194, 4.50497503], 
[3.68893535, 2.15916743, 2.38564121, 2.56312388, 2.18210999,
3.09240424, 1.1607601 , 3.73378146, 3.54388544, 2.46136629,
7.38952258, 5.92189193, 5.57668755, 5.45792671, 4.72537025,
5.08062988, 2.67719331, 4.12800194, 0.          , 4.25205833], 
[1.42702628, 3.85608143, 2.42937111, 5.949602 , 6.26426085,
7.23020913, 5.37783637, 7.69028504, 7.20011694, 5.27121276,
7.18618146, 4.51266706, 3.34395036, 2.63262683, 3.08007857,
3.22018633, 2.36142415, 4.50497503, 4.25205833, 0.          ]];

```

[]: ****
* OPL 22.1.0.0 Model
* Author: pardo

```

* Creation Date: 10 jun. 2022 at 8:35:21
*****/


//MODELO DE LA P-MEDIANA CON DISTANCIA MANHATTAN


// Defino las variables fijas
int n = ...;//número de cliente
int m = ...;//número de ubicaciones
int p=...;//número de instalaciones a ubicar

range cliente=1..n;
range ubicacion=1..m;
range dimension=1..2;

float u[cliente]=...;
float coor[cliente] [dimension]= ...;

float distancia[cliente] [ubicacion] ;

execute{
    for (var i in cliente){
        for(var j in ubicacion){
            //defino la distancia Manhattan
            distancia[i] [j]=Math.abs(coor[i] [1]-coor[j] [1])+Math.
            →abs(coor[i] [2]-coor[j] [2]);

        }
    }
}

// Defino variables de decision
dvar float+ x[cliente] [ubicacion];
dvar boolean y[ubicacion];

minimize sum (i in cliente,j in ubicacion) distancia[i] [j]*u[i]*x[i] [j];
subject to {
// se satisface la demanda de todos los clientes
forall (i in cliente) sum(j in ubicacion)
    x[i] [j] == 1;
//un cliente es atendido por una instalacion que esta abierta
forall (i in cliente, j in ubicacion)
    x[i] [j] <= y[j];
//se abren justamente p instalaciones
sum(j in ubicacion) y[j] == p ;
}

```

```

main {
    var f = new IloOplOutputFile("solucionpmedManhattan.txt");
    thisOplModel.generate();
    var time0 = new Date();
    var OK = cplex.solve();
    var time1 = new Date();

    var tTotal = (time1.getTime()-time0.getTime())/1000;

    if (OK) {
        f.writeln("Instalaciones abiertas");
        for(var j in thisOplModel.ubicacion){
            if(thisOplModel.y[j] != 0){
                f.writeln("Abro instalacion ",j);
            }
        }
        f.writeln("Asignacion de clientes");
        for(var j in thisOplModel.ubicacion){
            for(var i in thisOplModel.cliente){
                if(thisOplModel.x[i][j] != 0){
                    f.writeln("a la instalacion ",j," van a los clientes ",i);
                }
            }
        }
        f.writeln("Tiempo: ",tTotal,"segundos. Objetivo: "+cplex.getObjValue());
    }
    f.close();
}

```

```

[ ]: ****
* OPL 22.1.0.0 Data
* Author: Usuario
* Creation Date: 10 jun. 2022 at 8:35:21
****

//DATOS DE LOS MODELOS DE LA P-MEDIANA CON DISTANCIA MANHATTAN

//Numero de nodos
n=20;
m=20;
p=2;
// Se cambia p con el valor necesario en cada escenario

```

```

//Demanda en los puntos
u=[9,28,77,59,86,6,52,78,80,10,16,9,33,48,25,66,21,34,48,24];

//Coordenadas de los nodos
coor=[[0.8,4]
[3.46,4.2]
[2.14,3.64]
[5.66,4.02]
[6.06,2.7]
[7.02,2.56]
[5.12,2.18]
[6.68,-0.46]
[5.88,-0.88]
[3.88,-0.36]
[0.1,-4.2]
[-0.74,-1.5]
[-1.1,-0.24]
[-1.28,0.58]
[-0.22,-0.1]
[-0.558,-0.22]
[1.42,1.26]
[1.482,-1.2]
[3.962,2.1]
[-0.198,2.98]
];

```

3 Código y datos de los modelos de capacidad en IBM ILOG CPLEX Optimization Studio

```

[ ]: ****
* OPL 20.1.0.0 Model
* Author: Usuario
* Creation Date: 22 jun. 2022 at 18:06:25
****

//MODELOS CAPACIDAD CAP-U,CAP-M,CAP-U-C,CAP-U-C<600,CAP-M-RE,CAP-U-RE
//EN CADA ESCENARIO SE DESCOMENTA LA RESTRICCION NECESARIA Y SE EJECUTA

// Defino las variables fijas

```

```

int n = ...;//número de cliente

int m = ...;//número de ubicaciones

//int p=...; // número de instalaciones a ubicar

float eps=...;

range cliente=1..n;

range ubicacion=1..m;

range dimension=1..2;

float B[cliente]= ...;
float b[cliente]= ...;
float u[cliente]= ...; // numero de usuarios en cada centro de demanda
float coor[cliente][dimension]= ...;
float distancia[cliente][ubicacion]=...; // distancia entre los puntos
float f[ubicacion]=...;
float A[cliente][ubicacion];
float BB[cliente][ubicacion];
float C[cliente][ubicacion];

float d1;
float d2;

//conjuntos Pij

{int} P[i in cliente][j in ubicacion];

float cardinal[cliente][ubicacion];
//float coordenada[cliente][2];

//float demanda[cliente];

//float capacidad[ubicacion];

//execute initializeArray{

execute{
    var f = new IloOplOutputFile("datos.txt");
    var dr;

```

```

var d1;
var d2;
var aux1;
var aux2;
//calculo primero las ecuaciones de las rectas
for (var i in cliente){
    for(var j in ubicacion){
        //calculo la recta Ax+By+C que pasa por (xi,yi) y
        →(xj,yj) si i es distinto de j
        if(j != i){
            A[i][j]=coor[j][2]-coor[i][2];//  

            BB[i][j]=coor[i][1]-coor[j][1];//  

            C[i][j]=(coor[j][1]-
            coor[i][1])*coor[i][2]-(coor[j][2]-coor[i][2])*coor[i][1];
        →//
        }
        //recorro todos los clientes
        for (var k in cliente){
            dr=10;
            d1=10;
            d2=10;
            aux1=0;
            aux2=0
            //calculo la distancia del punto (xk,yk) a la
        →recta
            if (j != i) dr=Math.abs((A[i][j]*coor[k][1] +
            →BB[i][j]*coor[k][2] +
            C[i][j])/Math.sqrt(A[i][j]^2+BB[i][j]^2));
        →
            //calculo la distancia del punto (xk,yk) a (xi,yi)
            aux1=(coor[i][1]-coor[k][1])*(coor[i][1]-coor[k][1]);
            aux2=(coor[i][2]-coor[k][2])*(coor[i][2]-coor[k][2]);
            d1=Math.sqrt(aux1+aux2);
            //calculo la distancia del punto (xk,yk) a (xj,yj)
            aux1=(coor[j][1]-coor[k][1])*(coor[j][1]-coor[k][1]);
            aux2=(coor[j][2]-coor[k][2])*(coor[j][2]-coor[k][2]);
            d2=Math.sqrt(aux1+aux2);
            if(k==13 && i==2 && j== 2)
                f.writeln("aux1 y aux2 ",aux1, " y ", aux2,"d1 y d2",
            →",d1," ",d2);
                //Veo si está a menos de epsilon
                //if (Math.min(coor[i][1],coor[j][1])-eps <=
            →coor[k][1]
                //&& coor[k][1] <= Math.
            →max(coor[i][1],coor[j][1])+eps

```

```

//&& Math.abs((A[i][j]*coor[k][1] +_
→D[i][j]*coor[k][2] +
//C[i][j])/Math.sqrt((A[i][j])^2+(D[i][j])^2)) < eps)
if (dr<eps || d1<eps || d2 <eps)
{
    P[i][j].add(k);
    cardinal[i][j]=cardinal[i][j]+1; ↵
}
}
}
//fclose();
}

// Defino variables de decision

dvar float+ x[cliente][ubicacion];//o dvar boolean x[cliente][ubicacion]; si la_
→asignación es única

dvar boolean y[ubicacion];//si la ubicacion esta abierta

minimize sum (i in cliente , j in ubicacion) (distancia[i][j]*u[i]*x[i][j]) +_
→sum(j in ubicacion) f[j]*y[j];

subject to {

    //se atiende toda la demanda del cliente i

    forall (i in cliente)

        sum(j in ubicacion) x[i][j] == 1;

    //a un cliente solo se atiende desde ubicaciones abiertas

    forall (i in cliente , j in ubicacion)

        x[i][j] <= y[j];

    //cada ubicación atiende al menos la demanda mínima

    forall (j in ubicacion)

```

```

    sum(i in cliente) u[i]* x[i][j] >= b[j]*y[j];

    //cada ubicación atiende a lo mas la demanda máxima

    forall (j in ubicacion)

        sum(i in cliente) u[i]* x[i][j] <= B[j]*y[j] ;



        // los clientes van a la instalacion abierta mas cercana
        // forall(i in cliente , j in ubicacion)
            // sum(k in ubicacion : distancia[i][k] <= distancia[i][j]) x[i][k] >= y[j] ;



        //presupuesto de 600 para la apertura de instalaciones
        //sum(j in ubicacion) f[j]*y[j] <= 600 ;



        //todos los clientes cercanos a la ruta de i a j se asignen a la instalacion j

        //forall(i in cliente , j in ubicacion)

        //sum(k in P[i][j]) x[k][j] >= card(P[i][j]) * x[i][j] ;

    }

main {

    var f = new IloOplOutputFile("solucionCAP.txt");

    thisOplModel.generate();

    var time0 = new Date();

    var OK = cplex.solve();

    var time1 = new Date();




    var tTotal = (time1.getTime()-time0.getTime())/1000;
}

```

```

if (OK) {

    f.writeln("Instalaciones abiertas");

    for(var j in thisOplModel.ubicacion){

        if(thisOplModel.y[j] != 0){

            f.writeln("Abro instalacion ",j);

        }

    }

    f.writeln("Asignacion de clientes");

    for(var j in thisOplModel.ubicacion){

        for(var i in thisOplModel.cliente){

            if(thisOplModel.x[i][j] != 0){

                f.writeln("a la instalacion ",j," van\u2192los clientes ",i);

            }

        }

    }

    f.writeln("Tiempo: ",tTotal,"segundos. Objetivo: "+cplex.
    →getObjValue( ));

        for(var i in thisOplModel.cliente){
            for(var j in thisOplModel.ubicacion){
                //f.writeln("punto ",i,"valor x: ",thisOplModel.
    →coor[i][1],"valor y: ",thisOplModel.coor[i][2]);
                f.writeln("conjunto ",i,j,"cardinal :",thisOplModel.
    →cardinal[i][j]);
            }
        }
    }

    f.close();
}

```

```

[ ]: ****
* OPL 22.1.0.0 Data
* Author: Usuario
* Creation Date: 24 jun. 2022 at 14:55:37
****

//DATOS PARA LOS MODELOS CAP-U,CAP-M,CAP-U-C,CAP-U-C<600,CAP-M-RE,CAP-U-RE

//Número de nodos
n=20;
m=20;
eps=1.5;
//p=4;
//Coordenadas de los nodos
coor=[[0.8 4]
[3.46 4.2]
[2.14 3.64]
[5.66 4.02]
[6.06 2.7]
[7.02 2.56]
[5.12 2.18]
[6.68 -0.46]
[5.88 -0.88]
[3.88 -0.36]
[0.1 -4.2]
[-0.74 -1.5]
[-1.1 -0.24]
[-1.28 0.58]
[-0.22 -0.1]
[-0.558 -0.22]
[1.42 1.26]
[1.482 -1.2]
[3.962 2.1]
[-0.198 2.98]
];

//Demanda en los puntos
u=[9,28,77,59,86,6,52,78,80,10,16,9,33,48,25,66,21,34,48,24];

//Capacidades mínimas y máximas
B=[100,100,100,100,450,100,450,100,450,100,100,450,450,450,450,450,100,100,100];
→
b=[20,20,20,20,50,20,50,20,20,50,50,50,50,50,50,20,20,20,20];

```

```

//Costes fijos

f=[100,100,100,100,200,100,200,100,200,100,100,200,200,200,200,200,100,100,100] ;
→

// distancia entre los puntos
distancia=[[0, 2.6675082 , 1.38751577, 4.86004115, 5.41826541,
            6.38451251, 4.68772866, 7.3801084 , 7.04420329, 5.33816448,
            8.22982381, 5.71153219, 4.6462458 , 4.00284899, 4.22497337,
            4.43312125, 2.80927037, 5.24453277, 3.68893535, 1.42702628], ,
           [2.6675082 , 0.          , 1.43387587, 2.20735135, 3.0016662 ,
            3.91959182, 2.61457454, 5.664274 , 5.62697077, 4.57930126,
            9.04707688, 7.08025423, 6.36452669, 5.96422669, 5.65971731,
            5.97333441, 3.57843541, 5.75086811, 2.15916743, 3.85608143], ,
           [1.38751577, 1.43387587, 0.          , 3.54045195, 4.03112887,
            4.99807963, 3.31843337, 6.11731967, 5.86668561, 4.36206373,
            8.10106166, 5.89185879, 5.05489861, 4.58911756, 4.42235232,
            4.70943776, 2.48652368, 4.8845229 , 2.38564121, 2.42937111], ,
           [4.86004115, 2.20735135, 3.54045195, 0.          , 1.37927517,
            1.99529446, 1.91760267, 4.59464906, 4.90493629, 4.72787479,
            9.92380975, 8.45165073, 7.99031914, 7.74578595, 7.1797493 ,
            7.52602976, 5.05916989, 6.68611128, 2.56312388, 5.949602 ], ,
           [5.41826541, 3.0016662 , 4.03112887, 1.37927517, 0.          ,
            0.97015463, 1.07424392, 3.22024844, 3.58452228, 3.75712656,
            9.11765321, 7.99249648, 7.74010336, 7.64002618, 6.87592903,
            7.23355542, 4.85831246, 6.01399069, 2.18210999, 6.26426085], ,
           [6.38451251, 3.91959182, 4.99807963, 1.99529446, 0.97015463,
            0.          , 1.93762742, 3.03907881, 3.62397572, 4.28788992,
            9.67388236, 8.75792213, 8.58920252, 8.53290103, 7.71318352,
            8.07183275, 5.74891294, 6.69380639, 3.09240424, 7.23020913], ,
           [4.68772866, 2.61457454, 3.31843337, 1.91760267, 1.07424392,
            1.93762742, 0.          , 3.06646376, 3.15296686, 2.82651729,
            8.11817714, 6.91968207, 6.67418909, 6.596969 , 5.80637581,
            6.16438837, 3.81266311, 4.96582762, 1.1607601 , 5.37783637], ,
           [7.3801084 , 5.664274 , 6.11731967, 4.59464906, 3.22024844,
            3.03907881, 3.06646376, 0.          , 0.90354856, 2.80178515,
            7.56861942, 7.49252961, 7.78310992, 8.02765221, 6.90938492,
            7.24197791, 5.53407626, 5.25040989, 3.73378146, 7.69028504], ,
           [7.04420329, 5.62697077, 5.86668561, 4.90493629, 3.58452228,
            3.62397572, 3.15296686, 0.90354856, 0.          , 2.06649462,
            6.66564325, 6.64896985, 7.00927956, 7.30733878, 6.14966666,
            6.47174196, 4.94683737, 4.40962629, 3.54388544, 7.20011694], ,
           [5.33816448, 4.57930126, 4.36206373, 4.72787479, 3.75712656,
            4.28788992, 2.82651729, 2.80178515, 2.06649462, 0.          ,
            5.3883207 , 4.75857121, 4.98144557, 5.24492135, 4.10823563,
            4.44020765, 2.94550505, 2.54086678, 2.46136629, 5.27121276], ,

```

```

[8.22982381, 9.04707688, 8.10106166, 9.92380975, 9.11765321,
 9.67388236, 8.11817714, 7.56861942, 6.66564325, 5.3883207 ,
 0. , 2.8276492 , 4.13782552, 4.97521859, 4.11246884,
 4.03402578, 5.61729472, 3.30301741, 7.38952258, 7.18618146], ,
[5.71153219, 7.08025423, 5.89185879, 8.45165073, 7.99249648,
 8.75792213, 6.91968207, 7.49252961, 6.64896985, 4.75857121,
 2.8276492 , 0. , 1.31041978, 2.14895323, 1.49345238,
 1.29287432, 3.50473965, 2.24216057, 5.92189193, 4.51266706], ,
[4.6462458 , 6.36452669, 5.05489861, 7.99031914, 7.74010336,
 8.58920252, 6.67418909, 7.78310992, 7.00927956, 4.98144557,
 4.13782552, 1.31041978, 0. , 0.83952367, 0.89106678,
 0.54236888, 2.93264386, 2.75469127, 5.57668755, 3.34395036], ,
[4.00284899, 5.96422669, 4.58911756, 7.74578595, 7.64002618,
 8.53290103, 6.596969 , 8.02765221, 7.30733878, 5.24492135,
 4.97521859, 2.14895323, 0.83952367, 0. , 1.25936492,
 1.07762888, 2.7843132 , 3.28588557, 5.45792671, 2.63262683], ,
[4.22497337, 5.65971731, 4.42235232, 7.1797493 , 6.87592903,
 7.71318352, 5.80637581, 6.90938492, 6.14966666, 4.10823563,
 4.11246884, 1.49345238, 0.89106678, 1.25936492, 0. ,
 0.35866976, 2.13053984, 2.0265251 , 4.72537025, 3.08007857], ,
[4.43312125, 5.97333441, 4.70943776, 7.52602976, 7.23355542,
 8.07183275, 6.16438837, 7.24197791, 6.47174196, 4.44020765,
 4.03402578, 1.29287432, 0.54236888, 1.07762888, 0.35866976,
 0. , 2.47040159, 2.2631836 , 5.08062988, 3.22018633], ,
[2.80927037, 3.57843541, 2.48652368, 5.05916989, 4.85831246,
 5.74891294, 3.81266311, 5.53407626, 4.94683737, 2.94550505,
 5.61729472, 3.50473965, 2.93264386, 2.7843132 , 2.13053984,
 2.47040159, 0. , 2.46078118, 2.67719331, 2.36142415], ,
[5.24453277, 5.75086811, 4.8845229 , 6.68611128, 6.01399069,
 6.69380639, 4.96582762, 5.25040989, 4.40962629, 2.54086678,
 3.30301741, 2.24216057, 2.75469127, 3.28588557, 2.0265251 ,
 2.2631836 , 2.46078118, 0. , 4.12800194, 4.50497503], ,
[3.68893535, 2.15916743, 2.38564121, 2.56312388, 2.18210999,
 3.09240424, 1.1607601 , 3.73378146, 3.54388544, 2.46136629,
 7.38952258, 5.92189193, 5.57668755, 5.45792671, 4.72537025,
 5.08062988, 2.67719331, 4.12800194, 0. , 4.25205833], ,
[1.42702628, 3.85608143, 2.42937111, 5.949602 , 6.26426085,
 7.23020913, 5.37783637, 7.69028504, 7.20011694, 5.27121276,
 7.18618146, 4.51266706, 3.34395036, 2.63262683, 3.08007857,
 3.22018633, 2.36142415, 4.50497503, 4.25205833, 0. ]];

```

```

[ ]: ****
* OPL 20.1.0.0 Model
* Author: Usuario
* Creation Date: 22 jun. 2022 at 18:17:02
****/

```

```

//MODELOS CAP-U-RM Y CAP-M-RM

// Defino las variables fijas

int n = ...; //número de cliente
int m = ...; //número de ubicaciones
//int p=...; // número de instalaciones a ubicar
float eps=...;
range cliente=1..n;
range ubicacion=1..m;
range dimension=1..2;
float B[cliente]= ...;
float b[cliente]= ...;
float u[cliente]= ...; // numero de usuarios en cada centro de demanda
float coor[cliente][dimension]= ...;
float distancia[cliente][ubicacion]=...; // distancia entre los puntos
float f[ubicacion]=...;
float cardinal[cliente][ubicacion];

//conjuntos Pij

{int} P[i in cliente][j in ubicacion];



//float coordenada[cliente][2];
//float demanda[cliente];
//float capacidad[ubicacion];

//execute initializeArray{

execute{

    for (var i in cliente){

        for(var j in ubicacion){

            //defino la distancia Manhattan

            distancia[i][j]=Math.abs(coor[i][1]-coor[j][1])+Math.
→abs(coor[i][2]-coor[j][2]);

```

```

//defino los conjuntos (4 esquinas primero, rectangulo del
→medio, rectangulo de arriba, rectangulo de abajo,rectangulo izq, rectangulo
→derecho)

    for(var k in cliente){

        if (coor[k][1] <= Math.
→min(coor[i][1],coor[j][1])

            && coor[k][2] <= Math.
→min(coor[i][2],coor[j][2])

                && Math.abs(Math.
→min(coor[i][1],coor[j][1])-coor[k][1])+Math.abs(Math.
→min(coor[i][2],coor[j][2])-coor[k][2]) <= eps)
                    P[i][j].add(k);

    }

    for(var k in cliente){

        if (coor[k][1] <= Math.
→min(coor[i][1],coor[j][1])

            && coor[k][2] >= Math.
→max(coor[i][2],coor[j][2])

                && Math.abs(Math.
→min(coor[i][1],coor[j][1])-coor[k][1])+Math.abs(Math.
→max(coor[i][2],coor[j][2])-coor[k][2]) <= eps )
                    P[i][j].add(k);
    }

    for(var k in cliente){

        if (coor[k][1] >= Math.
→max(coor[i][1],coor[j][1])

            && coor[k][2] <= Math.
→min(coor[i][2],coor[j][2])

                && Math.abs(Math.
→max(coor[i][1],coor[j][1])-coor[k][1])+Math.abs(Math.
→min(coor[i][2],coor[j][2])-coor[k][2]) <=eps )

                    P[i][j].add(k);
    }
}

```

```

    }

    for(var k in cliente){

        if (coor[k][1] >= Math.
→max(coor[i][1],coor[j][1])

            && coor[k][2] >= Math.
→max(coor[i][2],coor[j][2])

            && Math.abs(Math.
→max(coor[i][1],coor[j][1])-coor[k][1])+Math.abs(Math.
→max(coor[i][2],coor[j][2])-coor[k][2]) <=eps )

            P[i][j].add(k);

    }

    for(var k in cliente){

        if (Math.min(coor[i][1],coor[j][1]) <=_
→coor[k][1]

            && coor[k][1] <= Math.
→max(coor[i][1],coor[j][1])

            && Math.min(coor[i][2],coor[j][2]) <= coor[k][2]

            && coor[k][2] <= Math.max(coor[i][2],coor[j][2]))

            P[i][j].add(k);

    }

    for(var k in cliente){

        if (Math.min(coor[i][1],coor[j][1]) <=_
→coor[k][1]

            && coor[k][1] <= Math.
→max(coor[i][1],coor[j][1])

            && Math.max(coor[i][2],coor[j][2]) <= coor[k][2]

            && coor[k][2] <= Math.max(coor[i][2],coor[j][2])+eps)

```

```

P[i][j].add(k);

}

for(var k in cliente){

    if (Math.min(coor[i][1],coor[j][1]) <= coor[k][1]
        && coor[k][1] <= Math.
        max(coor[i][1],coor[j][1])

        && Math.min(coor[i][2],coor[j][2]) - eps <= coor[k][2]
        && coor[k][2] <= Math.min(coor[i][2],coor[j][2]))


    P[i][j].add(k);

}

for(var k in cliente){

    if (Math.min(coor[i][1],coor[j][1]) - eps <= coor[k][1]
        && coor[k][1] <= Math.
        min(coor[i][1],coor[j][1])

        && Math.min(coor[i][2],coor[j][2]) <= coor[k][2]
        && coor[k][2] <= Math.max(coor[i][2],coor[j][2]))


    P[i][j].add(k);

}

for(var k in cliente){

    if (Math.max(coor[i][1],coor[j][1]) <= coor[k][1]
        && coor[k][1] <= Math.
        max(coor[i][1],coor[j][1]) + eps

        && Math.min(coor[i][2],coor[j][2]) <= coor[k][2]
        && coor[k][2] <= Math.max(coor[i][2],coor[j][2]))


    P[i][j].add(k);

}

```

```

        P[i][j].add(k);

    }
    cardinal[i][j]=opl.card(P[i][j]);
}

}

// Defino variables de decision

dvar float+ x[cliente][ubicacion];//o dvar boolean x[cliente][ubicacion]; si la
→asignación es única

dvar boolean y[ubicacion];//si la ubicacion esta abierta

minimize sum (i in cliente , j in ubicacion)  (distancia[i][j]*u[i]*x[i][j]) +_
→sum(j in ubicacion) f[j]*y[j];

subject to {

    //se atiende toda la demanda del cliente i

    forall (i in cliente)

        sum(j in ubicacion) x[i][j] == 1;

    //a un cliente solo se atiende desde ubicaciones abiertas

    forall (i in cliente , j in ubicacion)

        x[i][j] <= y[j];

    //cada ubicación atiende al menos la demanda mínima

    forall (j in ubicacion)

        sum(i in cliente) u[i]* x[i][j] >= b[j]*y[j];

    //cada ubicación atiende a lo mas la demanda máxima

```

```

forall (j in ubicacion)

    sum(i in cliente) u[i]* x[i][j] <= B[j]*y[j] ;



//todos los clientes cercanos a la ruta de i a j se asignen a la instalacion j

forall(i in cliente , j in ubicacion)

    sum(k in P[i][j]) x[k][j] >= card(P[i][j]) * x[i][j] ;

}

main {

    var f = new IloOplOutputFile("solucionCAP-M-R2.txt");

    thisOplModel.generate();

    var time0 = new Date();

    var OK = cplex.solve();

    var time1 = new Date();




    var tTotal = (time1.getTime()-time0.getTime())/1000;




    if (OK) {

        f.writeln("Instalaciones abiertas");

        for(var j in thisOplModel.ubicacion){

            if(thisOplModel.y[j] != 0){

                f.writeln("Abro instalacion ",j);

            }

        }

    }

}

```

```

        f.writeln("Asignacion de clientes");

        for(var j in thisOplModel.ubicacion){

            for(var i in thisOplModel.cliente){

                if(thisOplModel.x[i][j] != 0){

                    f.writeln("a la instalacion ",j," van a los clientes ",i);

                }

            }

        }

        f.writeln("Tiempo: ",tTotal,"segundos. Objetivo: "+cplex.getObjValue( ));

    }

    for(var i in thisOplModel.cliente){

        for(var j in thisOplModel.ubicacion){

            //f.writeln("punto ",i,"valor x: ",thisOplModel.coor[i][1],"valor y: ",thisOplModel.coor[i][2]);
            f.writeln("conjunto ",i,j,"cardinal :",thisOplModel.cardinal[i][j]);
        }

    }

    f.close();
}

}

```

```
[ ]: ****
* OPL 20.1.0.0 Data
* Author: Usuario
* Creation Date: 22 jun. 2022 at 18:17:02
****/
```

```
//DATOS DE LOS MODELOS CAP-U-RM Y CAP-M-RM
```

```
//Numero de nodos
n=20;
```

```

m=20;
eps=1.5;
//p=4;
//Coordenadas de los nodos
coor=[[0.8,4]
[3.46,4.2]
[2.14,3.64]
[5.66,4.02]
[6.06,2.7]
[7.02,2.56]
[5.12,2.18]
[6.68,-0.46]
[5.88,-0.88]
[3.88,-0.36]
[0.1,-4.2]
[-0.74,-1.5]
[-1.1,-0.24]
[-1.28,0.58]
[-0.22,-0.1]
[-0.558,-0.22]
[1.42,1.26]
[1.482,-1.2]
[3.962,2.1]
[-0.198,2.98]
];

//Demanda en los puntos
u=[9,28,77,59,86,6,52,78,80,10,16,9,33,48,25,66,21,34,48,24];

//Capacidades mínimas y máximas
B=[100,100,100,100,450,100,450,100,450,100,100,450,450,450,450,450,450,100,100,100,100];
→

b=[20,20,20,20,50,20,50,20,50,20,20,50,50,50,50,50,20,20,20,20];

//Costes fijos
f=[100,100,100,100,200,100,200,100,200,100,100,200,200,200,200,200,100,100,100,100];
→

```

4 Código y datos de los modelos jerarquizados en IBM ILOG CPLEX Optimization Studio

```
[ ]:/*****  
* OPL 22.1.0.0 Model  
* Author: Usuario  
* Creation Date: 24 jun. 2022 at 8:22:53  
*****/  
  
//ESCENARIOS DEL MODELO JERARQUIZADO  
  
  
// Defino las variables fijas  
int n = ...; // número de cliente  
int m = ...; // número de ubicaciones  
int s=...; // demanda  
  
range cliente=1..n;  
range ubicacion=1..m;  
range demanda=1..s;  
  
int B[cliente][demanda]= ...; // capacidades máximas  
int b[cliente][demanda]= ...; // capacidades mínimas  
int u[cliente][demanda]= ...; //numero de usuarios en cada centro de demanda  
float distancia[cliente][ubicacion] = ... ; //distancia entre los puntos  
float D[demanda]=...;  
float f[ubicacion][demanda]=...;  
int p[demanda]=...;  
  
  
// Defino variables de decision  
dvar boolean x[cliente][ubicacion][demanda];  
dvar boolean y[ubicacion][demanda];  
dvar float+ z[ubicacion][demanda][demanda];  
  
minimize sum (i in cliente , j in ubicacion , s in demanda) ↳  
→(distancia[i][j]*u[i][s]*x[i][j][s]) + sum(j in ubicacion , s in demanda) ↳  
→(f[j][s]*y[j][s]);  
  
subject to {  
  
forall (i in cliente , s in demanda)  
sum(j in ubicacion) x[i][j][s] == 1;
```

```

forall (i in cliente , j in ubicacion , s in demanda)
    x[i][j][s] <= sum(t in demanda : t >= s) y[j][t];

forall (j in ubicacion , s in demanda)
    sum(t in demanda : t >= s) z[j][s][t] == sum(i in cliente) u
    →u[i][s]*x[i][j][s];

forall (j in ubicacion , t in demanda)
    sum(s in demanda : s <= t) z[j][s][t] >= b[j][t]*y[j][t];

forall (j in ubicacion , t in demanda)
    sum(s in demanda : s <= t) z[j][s][t] <= B[j][t]*y[j][t];

forall (i in cliente , j in ubicacion , s in demanda)
    forall(t in demanda : t >= s)
        sum(k in ubicacion : distancia[i][k] <= distancia[i][j]) u
    →x[i][k][s] >= y[j][t];

forall(i in cliente, j in ubicacion, s in demanda : distancia[i][j] > D[s])
    x[i][j][s] == 0 ;

forall(s in demanda)
    sum(j in ubicacion) y[j][s] <= p[s];

}

main {
    var f = new IloOplOutputFile("solucionJER-0-(4,3).txt");
    thisOplModel.generate();
    var time0 = new Date();
    var OK = cplex.solve();
    var time1 = new Date();
    var tTotal = (time1.getTime()-time0.getTime())/1000;
    var instalacion1 = 0;
    var instalacion2 = 0;
    var instalacion3 = 0;
    var capacidad1 = 0;
    var capacidad2 = 0;
    var demanda1con2 = 0;

    if (OK) {
        f.writeln("Instalaciones abiertas");
        for(var j in thisOplModel.ubicacion){

```

```

        for(var s in thisOplModel.demanda){
            if(thisOplModel.y[j][s] != 0){
                f.writeln("Abro instalacion ", j , " de "
→tipo " , s);
            }
        }
    }

    f.writeln("Asignacion de clientes");
    for(var j in thisOplModel.ubicacion){
        for(var i in thisOplModel.cliente){
            for(var s in thisOplModel.demanda){
                if(thisOplModel.x[i][j][s] != 0){
                    f.writeln("a la instalacion ", j , " van "
→los clientes " , i , " de nivel " , s);
                }
            }
        }
    }

    f.writeln("Tiempo: " ,tTotal,"segundos. Objetivo: "+cplex.
→getObjValue( ));

    for(var j in thisOplModel.ubicacion){
        if(thisOplModel.y[j][1] != 0
→&& thisOplModel.y[j][2] != 0) instalacion3 = instalacion3+1;
        else{
            if(thisOplModel.y[j][1] != 0)
→instalacion1 = instalacion1+1;
            if(thisOplModel.y[j][2] != 0)
→instalacion2 = instalacion2+1;
        }
    }

    f.writeln("abierto 1: " , instalacion1);
    f.writeln("abierto 2: " , instalacion2);
    f.writeln("abierto1+2: " , instalacion3);
    for(var j in thisOplModel.ubicacion){
        if(thisOplModel.y[j][1] != 0) capacidad1 = capacidad1 +
→thisOplModel.B[j][1];
        if(thisOplModel.y[j][2] != 0) capacidad2 = capacidad2 +
→thisOplModel.B[j][2];
    }

    f.writeln("capacidad maxima nivel 1: " , capacidad1);
    f.writeln("capacidad maxima nivel 2: " , capacidad2);
    for(var j in thisOplModel.ubicacion){
        if(thisOplModel.z[j][1][2] != 0) demanda1con2 =
→demanda1con2 + thisOplModel.z[j][1][2];
    }
}

```

```

        f.writeln("demanda de nivel 1 con nivel 2: ", demanda1con2);
        f.close();
    }
}

```

```

[ ]: ****
* OPL 22.1.0.0 Data
* Author: Usuario
* Creation Date: 24 jun. 2022 at 8:22:53
****

//DATOS DE LOS MODELOS JERARQUIZADO
//Modificar p y D para resolver el escenario deseado


//Número de nodos
n=20;
m=20;
s=2;
p=[4,3];

//Tipos de instalacion (A,B,C o D):
→tipos=[A,C,A,B,B,A,D,C,C,D,A,A,A,A,D,D,B,C,D,D]

//Costes fijos costA=[100,500], costB=[250,1250], costC=[750,3750], costD=[1000,5000]
→costD=[1000,5000]

f=[[100,500],[750,3750],[100,500]
,[250,1250],[250,1250],[100,500]
,[1000,5000],[750,3750],[750,3750],[1000,5000]
,[100,500],[100,500]
,
→,[100,500],[100,500],[1000,5000],[1000,5000],[250,1250],[750,3750],[1000,5000],[1000,5000]];
→

//Demanda en los puntos (aleatoriedad dentro de cada tipo de instalación que ha tocado)
→tocado)
u=[[9, 1],[75, 30],[9, 1],[23, 8],[25, 7],[11, 1],[110, 45],[83, 33],[75,
→27],[110, 41],[10, 1],[5, 9],[11, 1],[11, 1],[100, 45],[90, 45],[23, 8],[68,
→30],[110, 41],[100, 50]];

```

```

//Capacidades mínimas y máximas

B=[[60,150],[450, 600],[60, 150],[150, 210],[150, 210],[60, 150],[600, 900],[450, 600],[450, 600],[600, 900],[60, 150],[60, 150],[60, 150],[60, 150],[600, 900],[600, 900],[150, 210],[450, 600],[600, 900],[600, 900]];

b=[[20, 50],[150, 200],[20, 50],[50, 70],[50, 70],[20, 50],[200, 300],[150, 200],[150, 200],[200, 300],[20, 50],[20, 50],[20, 50],[20, 50],[200, 300],[200, 300],[50, 70],[150, 200],[200, 300],[200, 300]];

//Distancia máxima a la instalación
D=[4,4];

//Distancia entre los nodos
distancia=[[0, 2.6675082 , 1.38751577, 4.86004115, 5.41826541,
            6.38451251, 4.68772866, 7.3801084 , 7.04420329, 5.33816448,
            8.22982381, 5.71153219, 4.6462458 , 4.00284899, 4.22497337,
            4.43312125, 2.80927037, 5.24453277, 3.68893535, 1.42702628],
           [2.6675082 , 0.          , 1.43387587, 2.20735135, 3.0016662 ,
            3.91959182, 2.61457454, 5.664274 , 5.62697077, 4.57930126,
            9.04707688, 7.08025423, 6.36452669, 5.96422669, 5.65971731,
            5.97333441, 3.57843541, 5.75086811, 2.15916743, 3.85608143],
           [1.38751577, 1.43387587, 0.          , 3.54045195, 4.03112887,
            4.99807963, 3.31843337, 6.11731967, 5.86668561, 4.36206373,
            8.10106166, 5.89185879, 5.05489861, 4.58911756, 4.42235232,
            4.70943776, 2.48652368, 4.8845229 , 2.38564121, 2.42937111],
           [4.86004115, 2.20735135, 3.54045195, 0.          , 1.37927517,
            1.99529446, 1.91760267, 4.59464906, 4.90493629, 4.72787479,
            9.92380975, 8.45165073, 7.99031914, 7.74578595, 7.1797493 ,
            7.52602976, 5.05916989, 6.68611128, 2.56312388, 5.949602 ],
           [5.41826541, 3.0016662 , 4.03112887, 1.37927517, 0.          ,
            0.97015463, 1.07424392, 3.22024844, 3.58452228, 3.75712656,
            9.11765321, 7.99249648, 7.74010336, 7.64002618, 6.87592903,
            7.23355542, 4.85831246, 6.01399069, 2.18210999, 6.26426085],
           [6.38451251, 3.91959182, 4.99807963, 1.99529446, 0.97015463,
            0.          , 1.93762742, 3.03907881, 3.62397572, 4.28788992,
            9.67388236, 8.75792213, 8.58920252, 8.53290103, 7.71318352,
            8.07183275, 5.74891294, 6.69380639, 3.09240424, 7.23020913],
           [4.68772866, 2.61457454, 3.31843337, 1.91760267, 1.07424392,
            0.          , 1.93762742, 3.03907881, 3.62397572, 4.28788992,
            9.67388236, 8.75792213, 8.58920252, 8.53290103, 7.71318352,
            8.07183275, 5.74891294, 6.69380639, 3.09240424, 7.23020913]];

```

1.93762742	0.	, 3.06646376	, 3.15296686	, 2.82651729
8.11817714	, 6.91968207	, 6.67418909	, 6.596969	, 5.80637581
6.16438837	, 3.81266311	, 4.96582762	, 1.1607601	, 5.37783637]
[7.3801084	, 5.664274	, 6.11731967	, 4.59464906	, 3.22024844
3.03907881	, 3.06646376	0.	, 0.90354856	, 2.80178515
7.56861942	, 7.49252961	, 7.78310992	, 8.02765221	, 6.90938492
7.24197791	, 5.53407626	, 5.25040989	, 3.73378146	, 7.69028504]
[7.04420329	, 5.62697077	, 5.86668561	, 4.90493629	, 3.58452228
3.62397572	, 3.15296686	, 0.90354856	0.	, 2.06649462
6.66564325	, 6.64896985	, 7.00927956	, 7.30733878	, 6.14966666
6.47174196	, 4.94683737	, 4.40962629	, 3.54388544	, 7.20011694]
[5.33816448	, 4.57930126	, 4.36206373	, 4.72787479	, 3.75712656
4.28788992	, 2.82651729	, 2.80178515	, 2.06649462	0.
5.3883207	, 4.75857121	, 4.98144557	, 5.24492135	, 4.10823563
4.44020765	, 2.94550505	, 2.54086678	, 2.46136629	, 5.27121276]
[8.22982381	, 9.04707688	, 8.10106166	, 9.92380975	, 9.11765321
9.67388236	, 8.11817714	, 7.56861942	, 6.66564325	, 5.3883207
0.	, 2.8276492	, 4.13782552	, 4.97521859	, 4.11246884
4.03402578	, 5.61729472	, 3.30301741	, 7.38952258	, 7.18618146]
[5.71153219	, 7.08025423	, 5.89185879	, 8.45165073	, 7.99249648
8.75792213	, 6.91968207	, 7.49252961	, 6.64896985	, 4.75857121
2.8276492	, 0.	, 1.31041978	, 2.14895323	, 1.49345238
1.29287432	, 3.50473965	, 2.24216057	, 5.92189193	, 4.51266706]
[4.6462458	, 6.36452669	, 5.05489861	, 7.99031914	, 7.74010336
8.58920252	, 6.67418909	, 7.78310992	, 7.00927956	, 4.98144557
4.13782552	, 1.31041978	0.	, 0.83952367	, 0.89106678
0.54236888	, 2.93264386	, 2.75469127	, 5.57668755	, 3.34395036]
[4.00284899	, 5.96422669	, 4.58911756	, 7.74578595	, 7.64002618
8.53290103	, 6.596969	, 8.02765221	, 7.30733878	, 5.24492135
4.97521859	, 2.14895323	, 0.83952367	0.	, 1.25936492
1.07762888	, 2.7843132	, 3.28588557	, 5.45792671	, 2.63262683]
[4.22497337	, 5.65971731	, 4.42235232	, 7.1797493	, 6.87592903
7.71318352	, 5.80637581	, 6.90938492	, 6.14966666	, 4.10823563
4.11246884	, 1.49345238	, 0.89106678	, 1.25936492	0.
0.35866976	, 2.13053984	, 2.0265251	, 4.72537025	, 3.08007857]
[4.43312125	, 5.97333441	, 4.70943776	, 7.52602976	, 7.23355542
8.07183275	, 6.16438837	, 7.24197791	, 6.47174196	, 4.44020765
4.03402578	, 1.29287432	, 0.54236888	, 1.07762888	, 0.35866976
0.	, 2.47040159	, 2.2631836	, 5.08062988	, 3.22018633]
[2.80927037	, 3.57843541	, 2.48652368	, 5.05916989	, 4.85831246
5.74891294	, 3.81266311	, 5.53407626	, 4.94683737	, 2.94550505
5.61729472	, 3.50473965	, 2.93264386	, 2.7843132	, 2.13053984
2.47040159	0.	, 2.46078118	, 2.67719331	, 2.36142415]
[5.24453277	, 5.75086811	, 4.8845229	, 6.68611128	, 6.01399069
6.69380639	, 4.96582762	, 5.25040989	, 4.40962629	, 2.54086678
3.30301741	, 2.24216057	, 2.75469127	, 3.28588557	, 2.0265251
2.2631836	, 2.46078118	0.	, 4.12800194	, 4.50497503]

```
[3.68893535, 2.15916743, 2.38564121, 2.56312388, 2.18210999,
 3.09240424, 1.1607601 , 3.73378146, 3.54388544, 2.46136629,
 7.38952258, 5.92189193, 5.57668755, 5.45792671, 4.72537025,
 5.08062988, 2.67719331, 4.12800194, 0.        , 4.25205833],  
[1.42702628, 3.85608143, 2.42937111, 5.949602 , 6.26426085,  
 7.23020913, 5.37783637, 7.69028504, 7.20011694, 5.27121276,  
 7.18618146, 4.51266706, 3.34395036, 2.63262683, 3.08007857,  
 3.22018633, 2.36142415, 4.50497503, 4.25205833, 0.        ]];
```

5 Cálculo de la distancia Euclídea entre los puntos con Python

```
[2]: import math
import numpy as np
```

```
[3]: p1=(0.8,4)
p2= (3.46,4.2)
p3= (2.14,3.64)
p4= (5.66,4.02)
p5= (6.06,2.7)
p6= (7.02,2.56)
p7= (5.12,2.18)
p8= (6.68,-0.46)
p9= (5.88,-0.88)
p10= (3.88,-0.36)
p11= (0.1,-4.2)
p12= (-0.74,-1.5)
p13= (-1.1,-0.24)
p14= (-1.28,0.58)
p15= (-0.22,-0.1)
p16= (-0.558,-0.22)
p17= (1.42,1.26)
p18= (1.482,-1.2)
p19= (3.962,2.1)
p20= (-0.198,2.98)
```

```
[4]: puntos=[p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16,p17,p18,p19,p20]
```

```
[5]: def distancia(punto1,punto2):
```

```
distancia = math.sqrt((punto2[0]-punto1[0])**2+(punto2[1]-punto1[1])**2)
return distancia
```

```
[6]: L=[]
for i in range(0,20):
    for j in range(0,20):
        distancias=distancia(puntos[i],puntos[j])

    L.append(distancias)
```

```
[7]: D=np.array(L).reshape(20,20)
D
```

```
[7]: array([[0.          , 2.6675082 , 1.38751577, 4.86004115, 5.41826541,
       6.38451251, 4.68772866, 7.3801084 , 7.04420329, 5.33816448,
       8.22982381, 5.71153219, 4.6462458 , 4.00284899, 4.22497337,
       4.43312125, 2.80927037, 5.24453277, 3.68893535, 1.42702628],
      [2.6675082 , 0.          , 1.43387587, 2.20735135, 3.0016662 ,
       3.91959182, 2.61457454, 5.664274 , 5.62697077, 4.57930126,
       9.04707688, 7.08025423, 6.36452669, 5.96422669, 5.65971731,
       5.97333441, 3.57843541, 5.75086811, 2.15916743, 3.85608143],
      [1.38751577, 1.43387587, 0.          , 3.54045195, 4.03112887,
       4.99807963, 3.31843337, 6.11731967, 5.86668561, 4.36206373,
       8.10106166, 5.89185879, 5.05489861, 4.58911756, 4.42235232,
       4.70943776, 2.48652368, 4.8845229 , 2.38564121, 2.42937111],
      [4.86004115, 2.20735135, 3.54045195, 0.          , 1.37927517,
       1.99529446, 1.91760267, 4.59464906, 4.90493629, 4.72787479,
       9.92380975, 8.45165073, 7.99031914, 7.74578595, 7.1797493 ,
       7.52602976, 5.05916989, 6.68611128, 2.56312388, 5.949602 ],
      [5.41826541, 3.0016662 , 4.03112887, 1.37927517, 0.          ,
       0.97015463, 1.07424392, 3.22024844, 3.58452228, 3.75712656,
       9.11765321, 7.99249648, 7.74010336, 7.64002618, 6.87592903,
       7.23355542, 4.85831246, 6.01399069, 2.18210999, 6.26426085],
      [6.38451251, 3.91959182, 4.99807963, 1.99529446, 0.97015463,
       0.          , 1.93762742, 3.03907881, 3.62397572, 4.28788992,
       9.67388236, 8.75792213, 8.58920252, 8.53290103, 7.71318352,
       8.07183275, 5.74891294, 6.69380639, 3.09240424, 7.23020913],
      [4.68772866, 2.61457454, 3.31843337, 1.91760267, 1.07424392,
       1.93762742, 0.          , 3.06646376, 3.15296686, 2.82651729,
       8.11817714, 6.91968207, 6.67418909, 6.596969 , 5.80637581,
       6.16438837, 3.81266311, 4.96582762, 1.1607601 , 5.37783637],
      [7.3801084 , 5.664274 , 6.11731967, 4.59464906, 3.22024844,
       3.03907881, 3.06646376, 0.          , 0.90354856, 2.80178515,
       7.56861942, 7.49252961, 7.78310992, 8.02765221, 6.90938492,
       7.24197791, 5.53407626, 5.25040989, 3.73378146, 7.69028504],
      [7.04420329, 5.62697077, 5.86668561, 4.90493629, 3.58452228,
       3.62397572, 3.15296686, 0.90354856, 0.          , 2.06649462,
```

6.66564325, 6.64896985, 7.00927956, 7.30733878, 6.14966666,
 6.47174196, 4.94683737, 4.40962629, 3.54388544, 7.20011694],
 [5.33816448, 4.57930126, 4.36206373, 4.72787479, 3.75712656,
 4.28788992, 2.82651729, 2.80178515, 2.06649462, 0. ,
 5.3883207 , 4.75857121, 4.98144557, 5.24492135, 4.10823563,
 4.44020765, 2.94550505, 2.54086678, 2.46136629, 5.27121276],
 [8.22982381, 9.04707688, 8.10106166, 9.92380975, 9.11765321,
 9.67388236, 8.11817714, 7.56861942, 6.66564325, 5.3883207 ,
 0. , 2.8276492 , 4.13782552, 4.97521859, 4.11246884,
 4.03402578, 5.61729472, 3.30301741, 7.38952258, 7.18618146],
 [5.71153219, 7.08025423, 5.89185879, 8.45165073, 7.99249648,
 8.75792213, 6.91968207, 7.49252961, 6.64896985, 4.75857121,
 2.8276492 , 0. , 1.31041978, 2.14895323, 1.49345238,
 1.29287432, 3.50473965, 2.24216057, 5.92189193, 4.51266706],
 [4.6462458 , 6.36452669, 5.05489861, 7.99031914, 7.74010336,
 8.58920252, 6.67418909, 7.78310992, 7.00927956, 4.98144557,
 4.13782552, 1.31041978, 0. , 0.83952367, 0.89106678,
 0.54236888, 2.93264386, 2.75469127, 5.57668755, 3.34395036],
 [4.00284899, 5.96422669, 4.58911756, 7.74578595, 7.64002618,
 8.53290103, 6.596969 , 8.02765221, 7.30733878, 5.24492135,
 4.97521859, 2.14895323, 0.83952367, 0. , 1.25936492,
 1.07762888, 2.7843132 , 3.28588557, 5.45792671, 2.63262683],
 [4.22497337, 5.65971731, 4.42235232, 7.1797493 , 6.87592903,
 7.71318352, 5.80637581, 6.90938492, 6.14966666, 4.10823563,
 4.11246884, 1.49345238, 0.89106678, 1.25936492, 0. ,
 0.35866976, 2.13053984, 2.0265251 , 4.72537025, 3.08007857],
 [4.43312125, 5.97333441, 4.70943776, 7.52602976, 7.23355542,
 8.07183275, 6.16438837, 7.24197791, 6.47174196, 4.44020765,
 4.03402578, 1.29287432, 0.54236888, 1.07762888, 0.35866976,
 0. , 2.47040159, 2.2631836 , 5.08062988, 3.22018633],
 [2.80927037, 3.57843541, 2.48652368, 5.05916989, 4.85831246,
 5.74891294, 3.81266311, 5.53407626, 4.94683737, 2.94550505,
 5.61729472, 3.50473965, 2.93264386, 2.7843132 , 2.13053984,
 2.47040159, 0. , 2.46078118, 2.67719331, 2.36142415],
 [5.24453277, 5.75086811, 4.8845229 , 6.68611128, 6.01399069,
 6.69380639, 4.96582762, 5.25040989, 4.40962629, 2.54086678,
 3.30301741, 2.24216057, 2.75469127, 3.28588557, 2.0265251 ,
 2.2631836 , 2.46078118, 0. , 4.12800194, 4.50497503],
 [3.68893535, 2.15916743, 2.38564121, 2.56312388, 2.18210999,
 3.09240424, 1.1607601 , 3.73378146, 3.54388544, 2.46136629,
 7.38952258, 5.92189193, 5.57668755, 5.45792671, 4.72537025,
 5.08062988, 2.67719331, 4.12800194, 0. , 4.25205833],
 [1.42702628, 3.85608143, 2.42937111, 5.949602 , 6.26426085,
 7.23020913, 5.37783637, 7.69028504, 7.20011694, 5.27121276,
 7.18618146, 4.51266706, 3.34395036, 2.63262683, 3.08007857,
 3.22018633, 2.36142415, 4.50497503, 4.25205833, 0.]])

6 Datos aleatorios para el problema jerarquizado con Python

```
[12]: import math  
import numpy as np  
import random
```

```
[20]: A=[[10,9,11],[1,0,2]]  
B=[[25,23,28],[8,7,9]]  
C=[[75,68,83],[30,27,33]]  
D=[[100,90,110],[45,41,50]]
```

```
[19]: costA=[100,500]  
costB=[250,1250]  
costC=[750,3750]  
costD=[1000,5000]
```

```
[14]: mMA=[[50,200],[150,500]]  
mMB=[[20,150],[60,300]]  
mMC=[[10,60],[50,250]]  
mMD=[[50,300],[200,600]]
```

```
[15]: tipos=[]
```

```
[16]: for i in range(0,20):  
    tipos.append(chr(random.randint(ord('A'), ord('D')))))
```

```
[17]: u=[[random.choice(A[0]),random.choice(A[1])],  
      [random.choice(C[0]),random.choice(C[1])],  
      [random.choice(A[0]),random.choice(A[1])],  
      [random.choice(B[0]),random.choice(B[1])],  
      [random.choice(B[0]),random.choice(B[1])],  
      [random.choice(A[0]),random.choice(A[1])],  
      [random.choice(D[0]),random.choice(D[1])],  
      [random.choice(C[0]),random.choice(C[1])],  
      [random.choice(C[0]),random.choice(C[1])],  
      [random.choice(D[0]),random.choice(D[1])],  
      [random.choice(A[0]),random.choice(A[1])],  
      [random.choice(B[0]),random.choice(B[1])],  
      [random.choice(A[0]),random.choice(A[1])],  
      [random.choice(D[0]),random.choice(D[1])],  
      [random.choice(D[0]),random.choice(D[1])],  
      [random.choice(B[0]),random.choice(B[1])],  
      [random.choice(C[0]),random.choice(C[1])],  
      [random.choice(D[0]),random.choice(D[1])],  
      [random.choice(D[0]),random.choice(D[1])]]
```

```
[21]: f=[costA,  
       costC,  
       costA,  
       costB,  
       costB,  
       costA,  
       costD,  
       costC,  
       costC,  
       costD,  
       costA,  
       costA,  
       costA,  
       costA,  
       costD,  
       costD,  
       costB,  
       costC,  
       costD,  
       costD]
```

```
[22]: b=[mMA[0] ,  
       mMC[0] ,  
       mMA[0] ,  
       mMB[0] ,  
       mMB[0] ,  
       mMA[0] ,  
       mMD[0] ,  
       mMC[0] ,  
       mMC[0] ,  
       mMD[0] ,  
       mMA[0] ,  
       mMA[0] ,  
       mMA[0] ,  
       mMD[0] ,  
       mMD[0] ,  
       mMB[0] ,  
       mMC[0] ,  
       mMD[0] ,  
       mMD[0]]
```

```
[23]: B=[mMA[1] ,  
       mMC[1] ,  
       mMA[1] ,  
       mMB[1] ,  
       mMB[1] ,
```

```
mMA[1],  
mMD[1],  
mMC[1],  
mMC[1],  
mMD[1],  
mMA[1],  
mMA[1],  
mMA[1],  
mMD[1],  
mMD[1],  
mMB[1],  
mMC[1],  
mMD[1],  
mMD[1]]
```

[]: